

Manual VMCLitio

D.J. Nader and B. M. Rubenstein

January 11th, 2023

This is an user manual for VMCLitio code which optimize compact explicitly-correlated wave functions for Lithium and isoelectronic sequence in the quartet state $1s2s2p$. Optimization for doublet state $1s^22p$ will be incorporated in this repository in the future. The aim of this manual is to provide a conceptual frame, introduce supplemental formulas and describe how this code was designed. The user is encouraged to modify the code in order to optimize explicitly-correlated wave functions for the study of different physical systems via VMC. Explicitly-correlated yet compact wave functions are useful in the study of quantum dots modeled by different confining potentials, three-atomic systems immersed in a dielectric media or interacting with external fields.

1 Supplemental formulas

Let us consider the following antisymmetric wave function

$$\Psi = \mathcal{A}\psi = \sum_P \hat{P}\psi,$$

where \mathcal{A} is the operator of antisymmetrization. For the computation of the local energy

$$E_L = \frac{\hat{H}\Psi}{\Psi},$$

one can use the following definition for the kinetic term

$$V_\psi \equiv \frac{\hat{T}\psi}{\psi} = -\frac{1}{2\psi} \sum_j^N \nabla_j^2 \psi, \quad (1)$$

where N indicates the number of electrons. Then the local energy is given by

$$E_L = \frac{1}{\Psi} \sum_P (-1)^P \hat{P}(\psi V_\psi) + V \quad (2)$$

where the sum stands for the permutations of the antisymmetric wave function and V is the potential term. Note that normalization is not necessary at this point since the local energy is given by the ratio. As for quantities $F_L^p = \frac{\partial_p \Psi}{\Psi}$ and $G_L^p = \frac{\hat{H} \partial_p \Psi}{\psi}$ necessary for the optimization, the derivatives of the wave function with respect to the variational parameters can be conveniently written in terms of the wave function

$$\frac{\partial \psi}{\partial p} = f_p(\mathbf{R})\psi(\mathbf{R}) \equiv \psi'_p(\mathbf{R}).$$

Thus, F_L^p can be calculated simply as

$$F_L^p = \frac{1}{\Psi} \sum_P (-1)^P \hat{P}(\psi'_p). \quad (3)$$

As for G_L^p it can be calculated as

$$G_L^p = \frac{\hat{T}}{\Psi} \sum_P (-1)^P \hat{P}(\psi'_p) + \frac{V}{\Psi} \sum_P (-1)^P \hat{P}(\psi'_p). \quad (4)$$

The second term is easy to treat since there are no derivatives. However the first term usually leads to complex formulas which may delay the computation time. Therefore it is convenient to reuse some quantities already calculated for local energy E_L . To this end we use the following property of the Laplacian operator

$$\nabla^2(uv) = u\nabla^2v + v\nabla^2u + 2\nabla u \cdot \nabla v.$$

Thus, the kinetic operator applying on the product $\psi'_p = f_p\psi$ can be calculated as

$$\hat{T}\psi'_p = -\frac{1}{2}\nabla^2(f_p\psi) = \psi'_p V_\psi + K_{\psi_p}, \quad (5)$$

defining the quantity

$$K_{\psi_p} = -\frac{1}{2}\psi\nabla^2 f_p - \sum_i^N \nabla_i \psi \cdot \nabla_i f_p,$$

where $\nabla^2 = \sum_{i=1}^N \nabla_i^2$. By substituting (5) in (4) we finally get

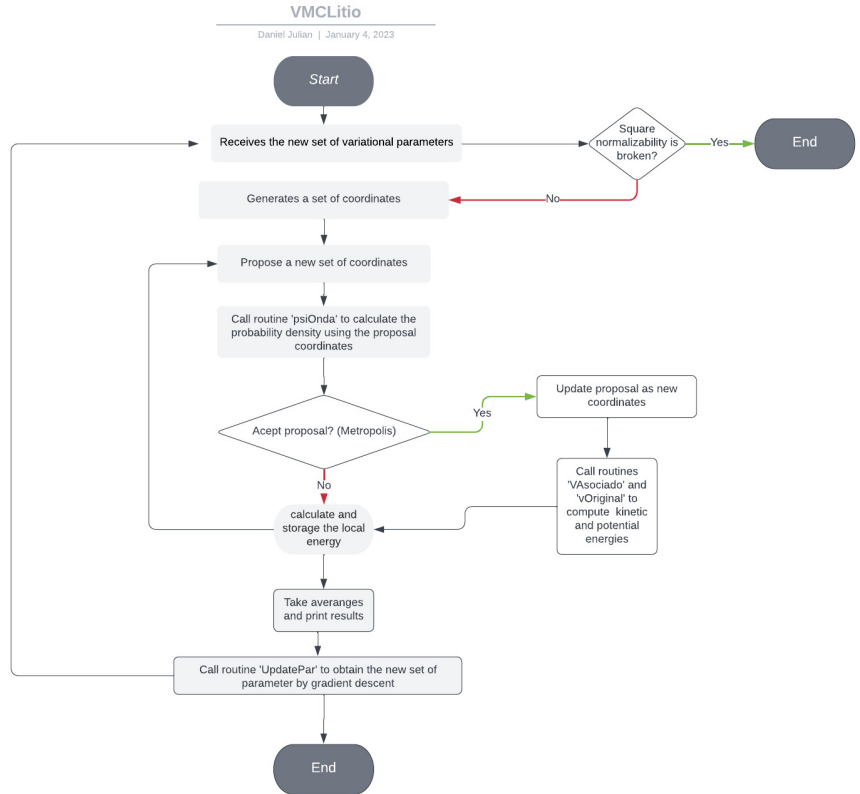
$$G_L^p = \frac{1}{\Psi} \sum_P (-1)^P \hat{P}(\psi'_p (V_\psi + V)) + \frac{1}{\Psi} \sum_P (-1)^P \hat{P}(K_{\psi_p}) \quad (6)$$

2 The algorithm

In practice, we wrote three subroutines in Fortran: i) 'psiOnda' which calculates ψ and permutations, ii) 'Vasociado' which calculates V_ψ and permutations, and finally iii) 'vOriginal' which calculates the potential of the system V . 'Vasociado' and 'Voriginal' are only called if the proposal of new coordinates is accepted in the Metropolis algorithm, thus reducing the computation time.

Quantities ψ'_p and K_{ψ_p} are computed within the routine 'Vasociado' for each permutation while (3) and (6) are calculated in a separated routine 'UPar' once all permutations are computed. Finally the updates of the variational parameters are calculated in the routine 'UpdatePar'.

The main routine in the code only read data from the input file and measure computation time. Therefore we present the flux diagram of the algorithm wich is performed in the routine FCNE



3 Using the code

In the inputfile one control the following options

1. Nuclear charge (Z) (double precision)
2. Maximum change in the coordinates (double precision): These variables which control the maximum change possible $\Delta \mathbf{R}$ in the electronic cylindrical coordinates during the algorithm of Metropolis from one step to the next. They are preferable adjusted in order to get 50 percent of acceptance in the Metropolis algorithm. This can be checked in the acceptance ratio printed on the output file.
3. The steps in the metropolis algorithm (N) (integer)
4. The Subiterations to estimate error (Ms) (integer): each iteration in the Metropolis algorithm is correlated to the previous one. We divide the total iterations N in Ms subgroups in order to estimate the statistical error. Ms must be tuned such that the ratio between Ms and the correlation time is larger than 100 in order to avoid underestimation of the statistical error. The correlation time (AC time) and the ratio between Ms and AC can be checked in the output file.

5. Iterations for the wave function optimization (k) (integer): The code will stop only because normalizability is broken or, in the ideal case, because it has reached the desired number of iterations. The convergence of the energy must be checked by following the evolution of the energy as a function of the iterations.
6. Lambda (double precision): corresponds to the parameter of the optimization λ_p . The code offer the option to set all λ_p equally for all parameters.
7. Variational paramaters: In the firs row, the number of variational parameters is set (integer). Then variational parameters are defined in three columns for i) name (string), ii) initial value (double precision) and iii) λ_p (double precision). For fixing/releasing parameters just fix λ_p as 0.0/1.0. Here, λ_p can also be set differently for each parameter in order to accelerate convergence.

We provide a sample Li.quartetPV3Z3.dat in the directory examples for optimizing Ansatz A in the case of neutral Lithium.

We recommend to use the linux command 'grep' to collect data from the output file in order to perform the data analysis.

4 Adaptation to different systems

In order to change the wave function, go to subroutine 'psiOnda' and replace variable psi(1). Consider that the subroutine receives cylindrical coordinates in the variables coor1, coor2 and coor3. The routine receives the values of variational parameters in the array varpars according to the order in the input file. The system of coordinates and names of variational parameters can be changed inside the routine. Conditions for square normalizability are set in the routine FCNE according to variables ANOR1, ANOR2 and ANOR3.

Correspondingly, the kinetic energy must be changed in the routine 'VASociado'. Same indications apply than those for routine 'psiOnda'. In this routine is necessary replace V_ψ in the variable Va(1). For simplicity, we divided the kinetic term in the contribution from the Laplacian of each electron in variables va1, va2 and va3 and we add them together in the variable Va(1). Also in this routine, the formulas for K_{ψ_p} and ψ_p are calculated and saved in the arrays aK and psip, respectively, for each variational parameter. The order of the list should match the order of parameters in the input file.

It is worthy to mention that subroutines 'psiOnda' and 'VASociado' only require to be modified for one single arbitrary permutation ψ of the wave function. The rest are done implicitly by permuting the coordinates to get the proper antisymmetric total wave function.

In order to change the potential go to routine 'vOriginal' and simply change the variable Vor.

The code use cylindrical coordinates by the foul. We recommend to preserve the cylindrical coordinates in the routine FCNE. However the conversion to more convenient coordinate system can be done in side of each subroutine.