

A SOFTWARE FOR GENERATING AND OPTIMIZING PSEUDOPOTENTIALS

R. Cimrman^{*}, J. Vackář^{**}, M. Novák^{***},

Abstract: *We wish to compute potentials for molecular dynamics simulations around a crack tip in iron that would be as precise as possible. The potentials will be constructed with help of our ab-initio electronic calculations code, that relies on the density functional theory and the pseudopotential approach. In this contribution we describe an approach and its new implementation for generating and optimizing the so called environment-reflecting all-electron pseudopotentials. A preliminary example is shown.*

Keywords: pseudopotentials, density functional theory, optimization.

1. Introduction

The electronic structure calculations represent a rigorous tool for predicting and understanding the properties of materials from first principles. Often, the calculations are based on the density functional theory (DFT) (Martin, 2005) combined with the pseudopotential approach (Vackář and Šimůnek, 2003). A pseudopotential is an operator that is used in place of the Coulombic potential of an atom core (which has a singularity at the atom centre) and has some properties that make the calculations feasible and more efficient. The accuracy, stability or speed of the calculations are influenced by the choice of the pseudopotential.

Our research in this field is motivated by the interest in a detailed modelling of cracks in iron, where we wish to compute potentials for molecular dynamics simulations around a crack tip that would be as precise as possible. The potentials will be constructed with help of our ab-initio electronic calculations code (Vackář et al., 2011; Cimrman et al., 2018), that relies on the pseudopotential approach.

This contribution is devoted to describing a new implementation of the approach developed in (Vackář, 1992) for generating and optimizing the so called environment-reflecting all-electron pseudopotentials. Our group had been using the original fortran 77 software written by J. Vackář for many years, but recently we decided to create a new code in Python to allow easier maintenance and enable future improvements, such as generalizing the procedure to heavy atoms, where relativistic effects are not negligible. The new software is a work-in-progress, so only a very limited example will be presented.

2. Density functional theory

In the density functional theory, the generally unsolvable many-particle Schrödinger equation is transformed to the Kohn-Sham equations (Kohn and Sham, 1965) — an eigenvalue problem with unknown eigenvalues (“energies”) ε_i and orbitals $\psi_i(\mathbf{r})$ (functions of space coordinates \mathbf{r}), in Hartree atomic units (as we use throughout the text):

$$\left(-\frac{1}{2}\nabla^2 + V_{\text{eff}}(\mathbf{r}, \rho) \right) \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r}), \quad (1)$$

where $\rho = \sum n_i |\psi_i|^2$ is the charge density and n_i are the occupation numbers. The charge density ρ computed in this way is the same that would solve the original Schrödinger equation, and thus it has a clear

^{*} Robert Cimrman, New Technologies - Research Centre, University of West Bohemia, cimrman3@ntc.zcu.cz

^{**} Jirí Vackář, Institute of Physics, Academy of Sciences of the Czech Republic, vackar@fzu.cz

^{***} Matyáš Novák, Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia & Institute of Physics, Academy of Sciences of the Czech Republic, novakmat@fzu.cz

physical interpretation, unlike $\psi_i(\mathbf{r})$. The total effective potential $V_{\text{eff}}(\mathbf{r}, \rho)$ can be written as

$$V_{\text{eff}}(\mathbf{r}, \rho) = V_{\text{ion}}(\mathbf{r}) + V_{\text{H}}(\mathbf{r}, \rho) + V_{\text{xc}}(\mathbf{r}, \rho) , \quad (2)$$

where $V_{\text{ion}}(\mathbf{r})$ is the ion potential, V_{xc} is the exchange-correlation potential describing the non-Coulomb electron-electron interactions and V_{H} is the electrostatic potential obtained as a solution to the Poisson equation $\nabla V_{\text{H}} = 4\pi\rho$. In a single atom, $V_{\text{ion}}(r)$ can be expressed in radial coordinates as $V_{\text{ion}}(r) = -Z/r$, where Z is the charge of the atom core. The singularity at the atom centre ($r = 0$) causes problems in numerical calculations and motivates replacing $V_{\text{ion}}(r)$ by a suitable pseudopotential.

3. Pseudopotential generation algorithm

Pseudopotentials can be generated for individual atoms, where the Schrödinger equation can be solved numerically in spherical coordinates. The radial part of the Schrödinger equation can be written as

$$\frac{\partial^2}{\partial r^2} (r\psi_{E,l}^{\text{xx}}) = \left(2(E - V^{\text{xx}}(r)) - \frac{l(l+1)}{r^2} \right) r\psi_{E,l}^{\text{xx}}(r) , \quad (3)$$

where \bullet^{xx} is either \bullet^{AE} (all-electron calculation) or \bullet^{PS} (pseudopotential generation), and l is the orbital quantum number.

In case of the all-electron calculation, (3) has a form of an eigenvalue problem for seeking the wave functions $\psi_{E,l}^{\text{AE}}(r)$ and the energies E . It can be solved for each l iteratively by the bisection method (bisecting the values of E to get a correct number of nodes of $\psi_{E,l}^{\text{AE}}(r)$) in combination with an ODE solver. In case of the pseudopotential generation, E is given and (3) is just an initial value problem, that can be solved by an ODE solver. In both cases, the initial conditions are specified using asymptotic values (Čertík et al., 2013).

A pseudopotential needs to meet several criteria, often mutually contradictory. Those include:

- no singularity at $r = 0$;
- problem size reduction: non-valence electrons are treated with the core;
- $\psi_{E,l}^{\text{PS}}(r) = \psi_{E,l}^{\text{AE}}(r)$ for $r \geq R_c$ with a given cut-off radius R_c ;
- smoothness at $r = R_c$, softness, transferability, etc. (Vackář and Šimůnek, 2003).

Here we outline the algorithm proposed in (Vackář, 1992), that allows generating semilocal (l -dependent) pseudopotentials as follows:

1. The screened pseudopotential $V_{E,l}^{\text{PS}}$ (i.e. including $V_{\text{H}} + V_{\text{xc}}$) is considered as a linear combination

$$V_{E,l}^{\text{PS}} = \sum_{j=0}^4 a_j F_j(r) , \quad (4)$$

where, for example,

$$\begin{aligned} F_0(r) &= 1 , & F_1(r) &= e^{-(\rho_1 r)^2} - 1 , & F_2(r) &= (\rho_2 r)^2 e^{-(\rho_2 r)^2} , \\ F_3(r) &= e^{-(\rho_3 r)^2} - 1 , & F_4(r) &= (\rho_4 r)^2 e^{-(\rho_4 r)^2} . \end{aligned} \quad (5)$$

It is generated for the given energy E and orbital quantum number l .

2. For a given cut-off radius R_c , the following conditions have to hold for the screened pseudopotential $V_{E,l}^{\text{PS}}$ and the all-electron potential V^{AE} :

$$V_{E,l}^{\text{PS}}(r) = V^{\text{AE}}(r) , \quad r \geq R_c , \quad (6)$$

$$\frac{\partial^i}{\partial r^i} V_{E,l}^{\text{PS}}(r) = \frac{\partial^i}{\partial r^i} V^{\text{AE}}(r) , \quad r = R_c , \quad i = 1, 2 . \quad (7)$$

3. Using normalization constants such that $\psi_{E,l}^{\text{PS}}(R_c) = \psi_{E,l}^{\text{AE}}(R_c)$, the following conditions hold:

$$\frac{\partial}{\partial r} \psi_{E,l}^{\text{PS}}(r) = \frac{\partial}{\partial r} \psi_{E,l}^{\text{AE}}(r), \quad r = R_c, \quad (8)$$

$$\int_0^{R_c} \left| \psi_{E,l}^{\text{PS}}(r) \right|^2 r^2 dr = \int_0^{R_c} \left| \psi_{E,l}^{\text{AE}}(r) \right|^2 r^2 dr. \quad (9)$$

The condition (9) corresponds to the charge conservation and makes the resulting pseudopotentials “norm-conserving”, which implies transferability, i.e. use at different energies. In (Vackář, 1992) the above conditions are replaced by a pair of equivalent conditions suitable for practical calculations due to an improved numerical stability.

3.1. The algorithm

The outline of the complete algorithm is as follows:

1. For each l , compute the all-electron potential $V^{\text{AE}}(r)$, radial wave functions $\psi_{E,l}^{\text{AE}}(r)$ and energies E using (3) with $\bullet^{\text{xx}} \equiv \bullet^{\text{AE}}$.
2. For given E, l , compute the screened pseudopotential $V_{E,l}^{\text{PS}}(r)$:
 - use a nonlinear solver to satisfy the conditions equivalent to (8)–(9) to obtain the coefficients a_0, a_4 ;
 - in each iteration of the nonlinear solver, compute $a_i, i = 1, 2, 3$ directly from (6),
 - after each change of an $a_i, i = 0, \dots, 4$, $\psi_{E,l}^{\text{PS}}(r)$ are computed using (3) with $\bullet^{\text{xx}} \equiv \bullet^{\text{PS}}$.
3. Unscreen $V_{E,l}^{\text{PS}}(r)$ by subtracting $V_{\text{xc}}(r, \rho(\psi_{E,l}^{\text{PS}}))$ and $V_{\text{H}}(r, \rho(\psi_{E,l}^{\text{PS}}))$.
4. Post-process into a form suitable for practical computations with a given software.

3.2. Optimization

The pseudopotential basis functions $F_j(r)$ in (4) can have several free parameters, such as $\rho_i, i = 1, 2, 3, 4$ in (5). Those parameters can be chosen in a way that is optimal w.r.t. some selected criterion, such as minimizing the “length” of Fourier image of the pseudopotential and of the corresponding wavefunction, minimizing the integrated curvature, minimizing the higher terms in Taylor expansion, and/or simply minimizing the depth of the pseudopotential. For this purpose, the algorithm 3.1. is embedded into an outer optimization loop. However, the relation between optimization criteria applied for an atom during pseudopotential construction and the resulting quality of the pseudopotential applied for the structure calculation is not known. The experience confirms that a simple “aesthetic” criterion (a nice smooth curve) belongs among very good ones, related to the results. In the example below we use the “softness” criterion:

$$\left| \frac{\partial^2}{\partial r^2} V_{E,l}^{\text{PS}}(r) \right|_{r=0} = \min. \quad (10)$$

4. Software and a preliminary example

Our new software for generating and optimizing pseudopotentials is currently under development. The logic of the application is written in Python to allow an easy maintenance and future enhancements.

The main script allows full control over the computation using a command line interface. This allows an easy algorithm selection and exploratory calculations. Various components of the calculation, such as the grids, solvers, formulas, or visualizations, are implemented as plugins that can be simply added, modified or removed. For the optimization and root finding, the solvers from SciPy (Jones et al., 2019) can be used.

As a very preliminary example we show a result concerning the pseudopotential generation and optimization for the nitrogen atom. This atom has 7 electrons in three sub-shells 1s: 2, 2s: 2, 2p: 3, with energies $E_{1s2} \approx -14.01$, $E_{2s2} \approx -0.676$, $E_{1s2} \approx -0.266$. Note that the exact values depend on the choice of V_{xc} and other considerations. In Fig. 1 a comparison of $V^{\text{AE}}(r)$ with $V_{E,l}^{\text{PS}}(r)$ is shown, for the energy $E \approx -0.676$ and $l = 0$. We can see that the “optimized” (w.r.t. (10)) pseudopotential V_1^{PS} is shallower when compared to the initial V_0^{PS} , and the curve is nice and smooth, adhering to the “aesthetic” criterion.

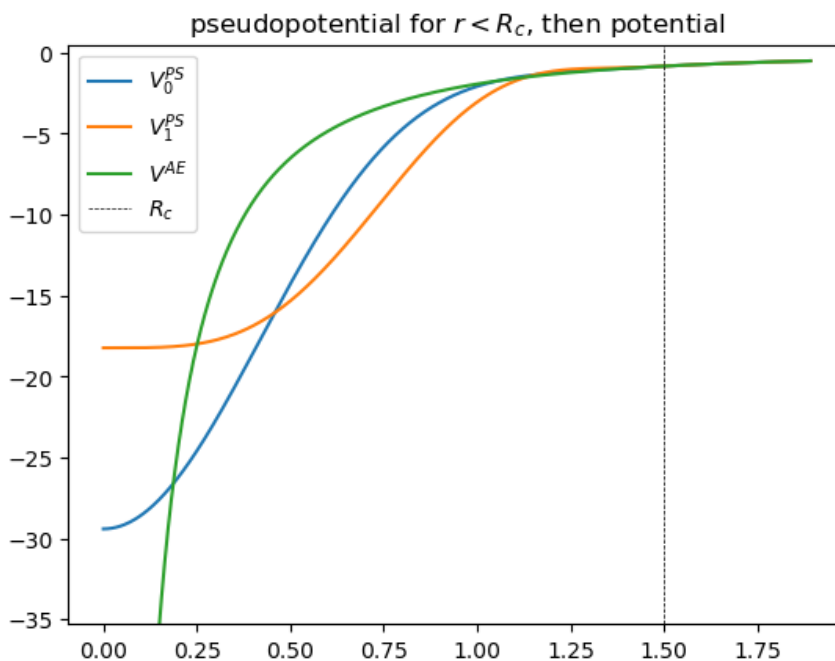


Fig. 1: The comparison of $V^{\text{AE}}(r)$ with $V_{E,l}^{\text{PS}}(r)$ for the nitrogen atom, $E \approx -0.676$, $l = 0$. The V_0^{PS} curve corresponds to the initial values of the optimization parameters ρ_i , while V_1^{PS} to the final values.

5. Conclusion

We described some aspects of generating and optimizing pseudopotentials in density functional theory based electronic structure calculations and presented a preliminary example calculated with our new Python implementation. The code will be further improved to work also with the relativistic Dirac equation, enabling calculations with heavy atoms. Ultimately, accurate site-specific potentials for molecular dynamics simulations of a crack tip in iron will be generated with the help of this code in combination with our electronic structure calculations code.

Acknowledgments

The work was supported by the Czech Science Foundation, grant projects GA17-12925S (the first and second authors) and GA17-14840S (the third author).

References

- Cimrman, R., Novák, M., Kolman, R., Tůma, M., Plešek, J., and Vackář, J. (2018). Convergence study of isogeometric analysis based on Bézier extraction in electronic structure calculations. *Applied Mathematics and Computation*, 319:138 – 152.
- Jones, E., Oliphant, T., Peterson, P., et al. (2001–2019). SciPy: Open source scientific tools for Python. [Online; accessed 2019-02-14].
- Kohn, W. and Sham, L. J. (1965). Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138. doi:10.1103/PhysRev.140.A1133.
- Martin, R. M. (2005). *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press.
- Vackář, J. and Šimůnek, A. (2003). Adaptability and accuracy of all-electron pseudopotentials. *Phys. Rev. B*, 67. 125113.
- Vackář, J. (1992). Konstrukce zobecněných semilokálních normu-zachovávajících pseudopotenciálů přímou metodou. Kandidátská disertační práce, Fyzikální ústav Československé akademie věd.
- Vackář, J., Čertík, O., Cimrman, R., Novák, M., Šipr, O., and Plešek, J. (2011). *Advances in the Theory of Quantum Systems in Chemistry and Physics*, volume 22 of *Prog. Theoretical Chem. and Phys.*, chapter Finite Element Method in Density Functional Theory Electronic Structure Calculations, pages 199–217. Springer.
- Čertík, O., Pask, J. E., and Vackář, J. (2013). dftatom: A robust and general Schrödinger and Dirac solver for atomic structure calculations. *Computer Physics Communications*, 184(7):1777–1791.