"Science Stays True Here"
Biological and Chemical Research, Volume 2015, 1-10 | Science Signpost Publishing

Symbolic Regression of Inter-Atomic Potentials *via* Genetic Programming

Abdel Kenoufi[1], Kholmirzo T. Kholmurodov[2]

1. Scientific Consulting for Research and Engineering, Strasbourg, France.

2. Frank Laboratory of Neutron Physics, Dubna, Moscow, Russia.

Received: October 13, 2014 / Accepted: November 11, 2014 / Published: January 25, 2015

Abstract: One presents methodology and algorithm to find inter-atomic potentials for different kinds of material systems. One uses *ab initio* potential energy calculations and symbolic regression using genetic programming, which is a generalization of genetic algorithms.

Keywords: Inter-atomic potentials, Molecular Dynamics, *Ab Initio* calculations, Density Functional Theory, Kohn-Sham scheme, Evolutionary Programming, Symbolic Regression

1. Introduction

Many research in nuclear sciences and technologies are focused on aging of nuclear reactors and vessels, or on biological molecules damages [1]. In both cases, nuclear engineers and scientists are taking benefit from the fast evolution of powerful computing facilities to develop simulation tools in order to complete missing informations from experimentations. One of the main challenge is then multi-scale simulations with coupling of codes working at different scales, microscopic (*ab initio*), mesoscopic(molecular dynamics), macroscopic (structure calculations with finite elements) [1]. Usually, material physicists are using molecular dynamic codes for calculations on alloys in order to determine thermo-mechanical constants, such as elastic constants, Young modulus, ... To achieve that, those codes need knowledge of well-suited inter-atomic potentials for each type of material system (molecule, alloy, ...), which yields to a long and very difficult development. Moreover, no mathematical shape can be speculated for binary and ternary alloys. One presents in the two first sections a systematic procedure using input/output data coming from experiments. It uses symbolic regression based on genetic programming [2]. In the last section it is applied to material systems with experimental data and/or *ab initio* calculations for determining an analytical formula for inter-atomic potential. One exhibits some examples with numerical applications developed in *ruby* which is an elegant, powerful object oriented programming language [10].

2. Symbolic Regression *via* Genetic Programming (SR*v*GP)

2.1. Scheme

Usually if some input-output data are known, one performs function parameters fitting using optimization methods. Because it needs some knowledge of an empirical and parametrized form of the function, and since it is hard for the

Corresponding author:

Abdel Kenoufi, Scientific Consulting for Research and Engineering, Strasbourg, France. E-mail: kenoufi@s-core.fr.

numerical scheme not to be trapped in a local minimum, it can be rather useful and powerful to perform a symbolic regression in order to find a function converging uniformly to the unknown one as shown on figure (1).

Symbolic regression is a good candidate to achieve such kind of procedures. It has the great advantage to give a formal expression of the wanted function which converges uniformly on the adjustment set (figure 1) to the unknown one. Uniform convergence means that it is stronger than point-wise convergence and that the speed of convergence is not point dependant.

Symbolic regression can be done in different ways. In this article, we focus on the so-called Symbolic Regression *via* Genetic Programming (GP) [4-6] (*SRvGP*). GP was initially developed for inductive automatic programming and is well-suited for symbolic regression, controller design, and machine learning tasks. One can consider GP algorithms as an extension of the well-known genetics algorithms. GP's scope is to use induction to devise a computer program. This isachieved by using evolutionary operators on candidate programs with a tree structure to improve the adaptive fit between the population of candidate programs and an objective function. An assessment of a candidate solution involves its execution. Symbolic regression with input-output data is achieved just by considering a function as a computer program, which is represented with a tree (figure 3) and to use those data in the objective function.

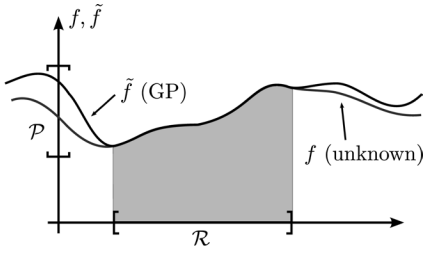

Fig. 1. Symbolic Regression *via* Genetic Programming permits to build an approximation of an unknown function defined on $\mathcal{R}$, and valued on $\mathcal{P}$.

Let us suppose that some input-output data ($\mathcal{R}$, $\mathcal{P}$) have been measured where $\mathcal{R} \subset \mathbb{R}^n$ and $\mathcal{P} \subset \mathbb{R}^m$ and $n, m \in \mathbb{N}^*$. Therefore, we would like to find out a symbolic formula which can describe the model by mean ofthe symbolic regression using a predefined set of mathematical and logical operators such as for instance {+, −, ·, /, exp, log, sin, cos, neg} where neg: $x \mapsto -x$. One can use unary, binary, ..., or any n-ary operator. It can be useful sometimes to perform pre-treatments of the data. If the order of magnitude of input-output data are very different, it is necessary to normalize them with the maximal values in order to prevent crushing of the largest values on the smallest and to keep homogeneous order of magnitudes: $\mathcal{R}_i \mapsto \frac{\mathcal{R}_i}{\mathcal{R}_{i,\max}}$ for $1 \le i \le n$, and $\mathcal{P}_i \mapsto \frac{\mathcal{P}_i}{\mathcal{P}_{i,\max}}$ for $1 \le i \le m$. The symbolic regression is then performed on this new data set and the right model is recovered with a re-normalization as a post-treatment process. Another pre-treatment which can be done is to remove noise from output data with Fourier or wavelets transforms [3]. This signal processing needs that one has to choose the frequencies which have to been kept or removed. On the other hand, it can have meaning to find a solution which takes into account the noise, but this is another approach which is not discussed in this paper and can be studied in another framework. In a second step, a decimation of the output data can be useful to decrease the computational time and to avoid over-fitting phenomena. Wavelet analysis seems to be a promising tool [3] to achieve that. Finally, one obtains a new input-output data set which can be used for the symbolic regression of the model.

3. Mathematical Example of SR*v*GP

For example, we propose to illustrate this procedure on data produced with the function:

$$f(x) = \sin(5x) \cdot e^{-x^2} \text{ for } x \in [-3, 3] \tag{1}$$

This signal is very interesting, not only because it is a wavelet [3], but because it has several local extrema and fast oscillations.

We have added to this function a random uniformly distributed noise $\varepsilon(x) \in [-0.25, 0.25]$ for $x \in [-3, 3]$ in order to see if, in this case, the genetic programming scheme is sensible to small perturbations or could extract the right signal we are looking for. In this example, one has allowed only random generation of integer numbers as leafs of the tree.

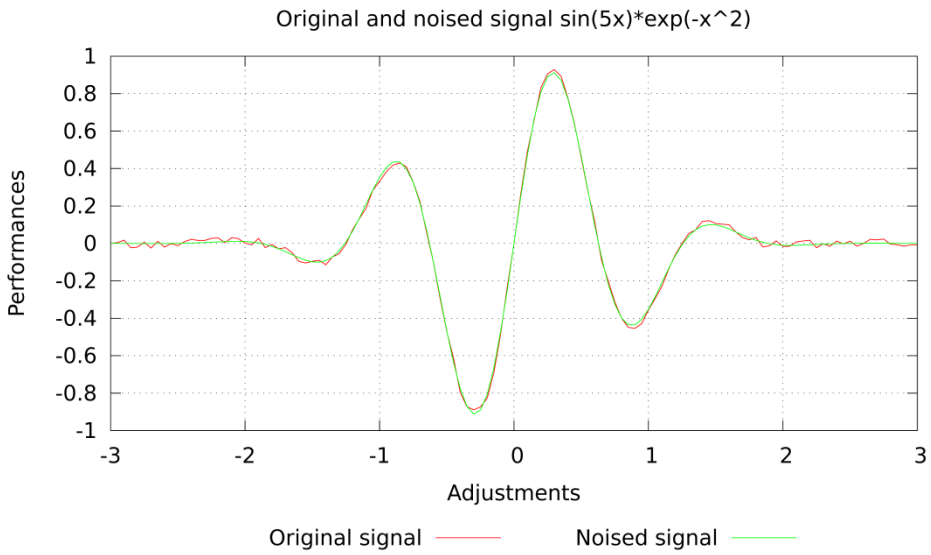

Fig. 2. Graphical representation of the function $f(x) = \sin(5 \cdot x) \cdot e^{-x^2}$. An uniform random noise $\varepsilon(x) \in [-0.25, 0.25]$ for $x \in [-3, 3]$ has been added to the initial output data set *f*([-3,3]).

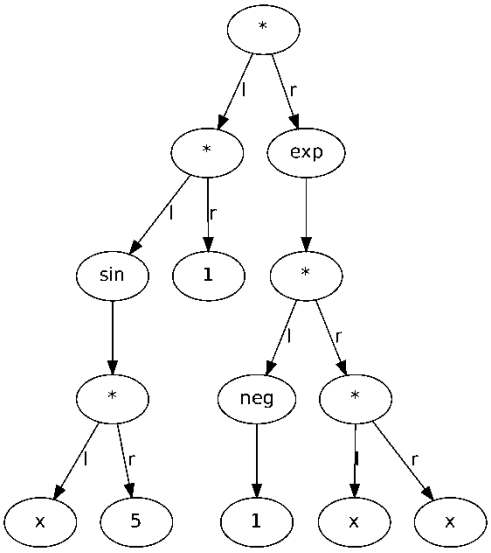

Fig. 3. Result for the genetic programming of the $f(x) = \sin(5x) \cdot e^{-x^2}$ noised example with integer random numbers generation. This tree corresponds to the approximation of f, $\tilde{f}(x) = \sin(5x) \cdot e^{-x^2} = f(x)$ on $[-3, 3]$ with function basis set {+, −, ·, /, exp, log, sin, cos, tan, neg}. The letters "l" and "r" correspond respectively to left and right positions of operands in a binary expression.

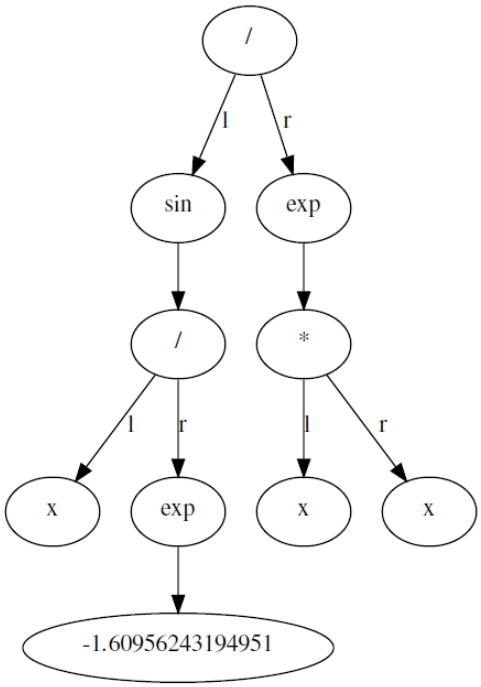

Fig. 4. Result for the genetic programming of the $f(x) = \sin(5x) \cdot e^{-x^2}$ noised example with real numbers random generation.

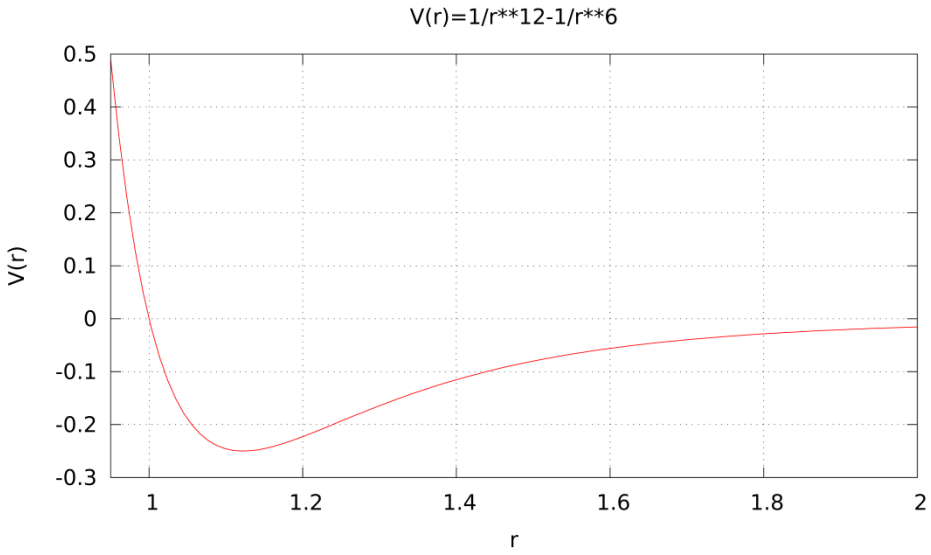

Fig. 5. Lennard-Jones-like potential example.

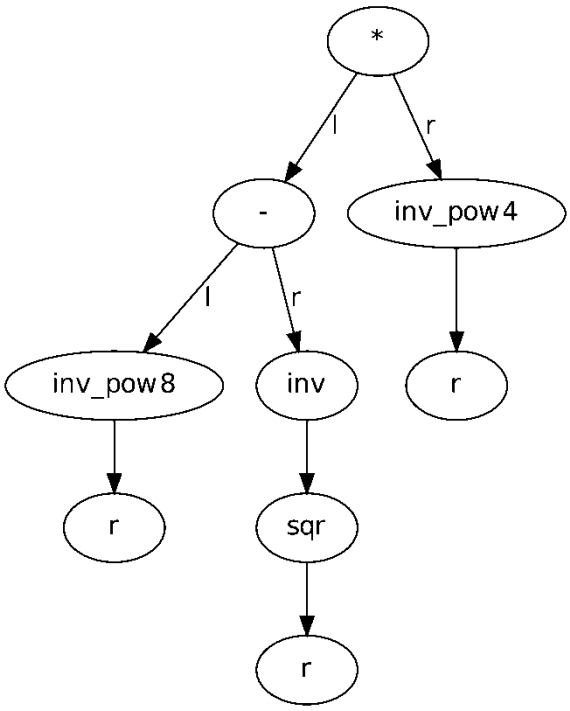

Fig. 6. SR*v*GP solution for Lennard-Jones-like potential. The right solution is found in few hundred iterations: $V(r) = \frac{1}{r^{12}} - \frac{1}{r^6}$

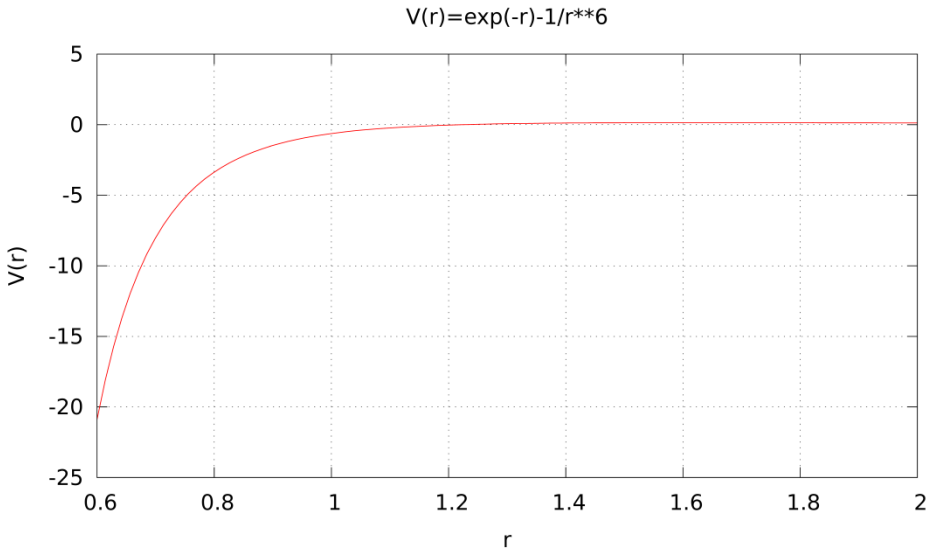

Fig. 7. Buckingham-like potential example.

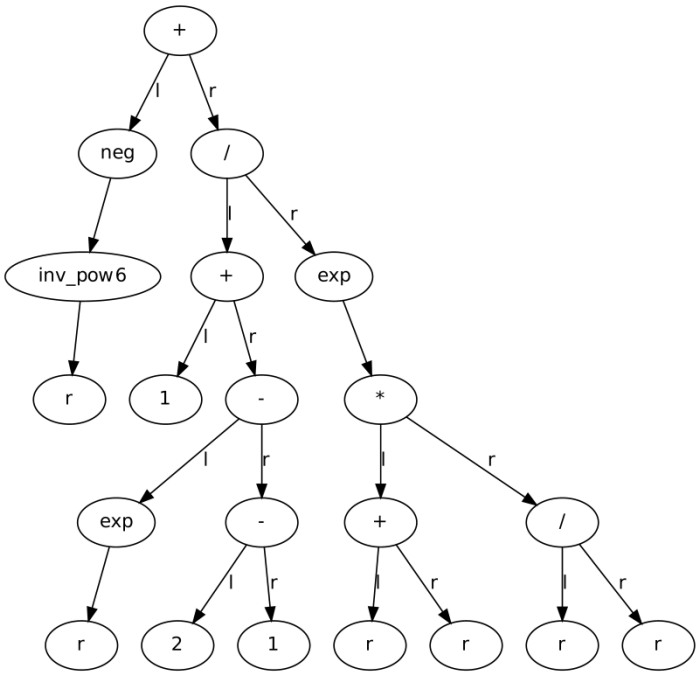

Fig. 8. SR*v*GP solution for Buckingam-like potential. The right solution is found in few hundred iterations: $V(r) = e^{-r} - \frac{1}{r^6}$

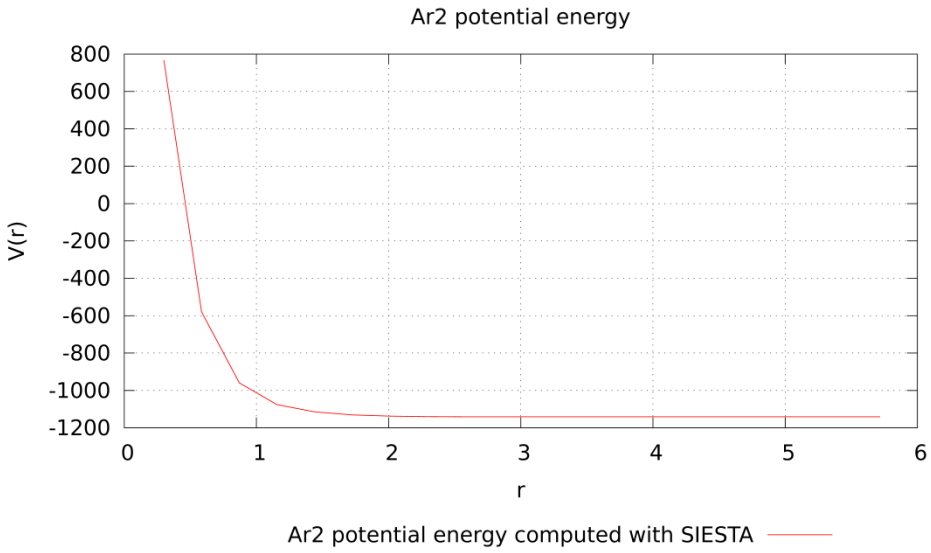

Fig. 9. Ar_2 molecule potential energy computed with *SIESTA*.

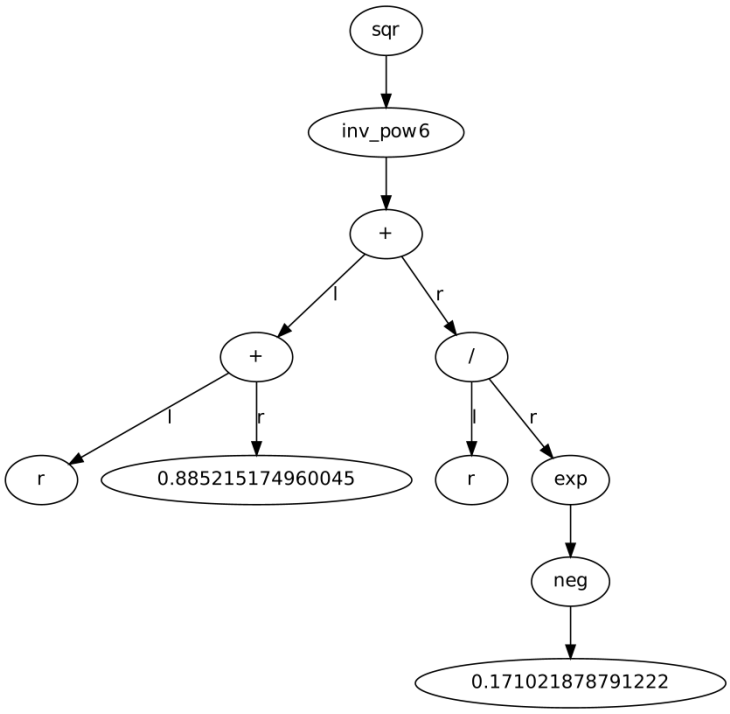

Fig. 10. SR*v*GP solution for Ar_2 molecule potential energy computed with *SIESTA*. The right solution is found in few thousand iterations for a fitness less than $6 \cdot 10^{-4}$: $V(r) = [r \cdot (1 + e^{0.171021878791222}) + 0.885215174960045]^{-12}$

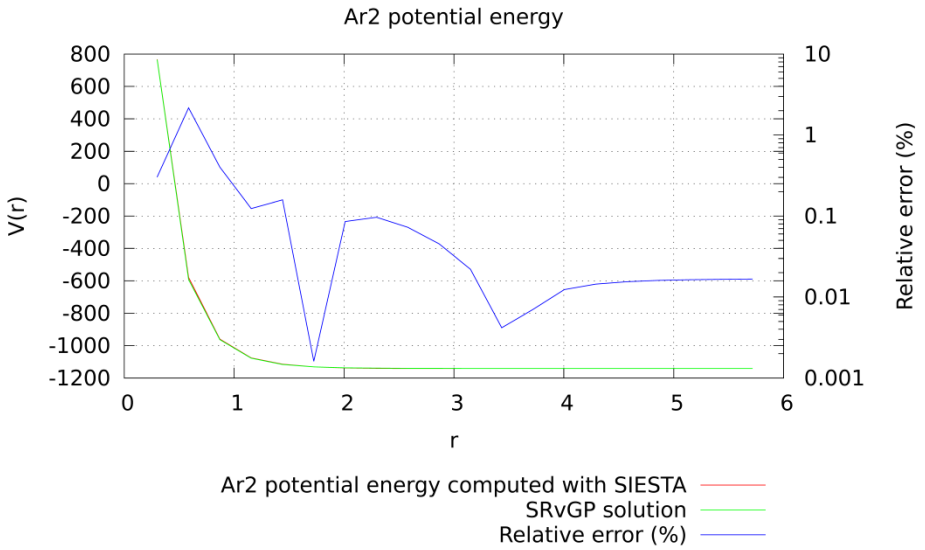

Fig. 11. Representation of the SR*v*GP solution after renormalization $V(r) = 1906.22 \cdot [\frac{r}{5.71} \cdot (1 + e^{0.171021878791222}) + 0.885215174960045]^{-12} - 1141.370159$ and the potentiel energy computed with *SIESTA* DFT code.

The symbolic regression code [7] has been implemented within a *ruby* script [10] and a MPI library [9] for this language which has been developed [8] especially for the scope of this present article. The code is well-suited for parallel computing, launching several jobs on different nodes with different random seeds. After convergence of each population, a last run is done on a single node involving the best candidates of each previous nodes within a larger population. The *ruby* code is also able to produce automatically the derivatives of the function found with SR*v*GP.

The final result obtained in less than 3 minutes for this example with 16 processors and for a fitness function less than 10^{-5}, is given in a *EBNF* [11] (Extended Backus-Naur Form) expression (*(*(sin(*x5))1)(exp(*(neg1)(*xx)))) and can be represented by the tree shown on figure (3). It represents the function:

$$\tilde{f}(x) = \sin(5x) \cdot e^{-x^2} \tag{2}$$

We have choosen a grow method for the tree development with a maximal depth of 5 and a set of function such as {+, −, ·, /, exp, log, sin, cos, tan, neg} where $\mathrm{neg}(x) = -x$. Figure (2) shows the noised signals for input on [−3, 3].

It is interesting that in this example the signal filtering has not been used in this example since the genetic program absorbs the noise and finds the right solution. Thus, the symbolic regression finds finally the right function $\tilde{f} = f$ on the input set [−3, 3] even in presence of small noise and without filtering. This means that the model is very robust and not sensible to small numerical or experimental noises in general.

It is interesting to see what happens if one permits to generate not only integer numbers, but real ones, as it is exhibited on figure (4). This final result obtained in 5 minutes for this example with 2 processors (2 Ghz, Intel Dual Core, 4 Gb RAM) is given in a *EBNF* [11] expression like (/(sin(/x(exp−1.60956243194951)))(exp(∗xx))) and can be represented by the tree shown on figure (4). This expression represents the function:

$$\tilde{f}(x) = \sin(e^{1.60956243194951} \cdot x) \cdot e^{-x^2} \tag{3}$$

Since $e^{1.60956243194951} \simeq 5.000622636341432$, the symbolic regression converges finally to the right function $\tilde{f} \simeq f$ on the input set [−4, 3].

4. Applications for Material Systems

The main scope of this section is to apply SR*v*GP to data coming from experiments or *ab initio* simulations in order to find for example inter-atomic potentials, chemical, thermodynamic and mechanical constants which can be used to transfer informations from one space-time scale to a higher one.

We show in this section some simple but relevant examples in order to exhibit the feasibility of the SR*v*GP scheme.

4.1. Empirical inter-atomic potentials

It is interesting to see if SR*v*GP is efficient for interatomic potentials such as Lennard-Jones, so-called 6-12 potential (Figure 5) and Born-Mayer like potentials, 6-exp potential (Figure 7). We use reduced units to perform calculations and to show how SR*v*GP is efficient and converges to the right results. The algorithm uses the input-output data given through the analytic form of the potentials. And then, one applies SR*v*GP to those data for a maximal depth of 6, with $\{+, -, \cdot, /, \exp, \mathrm{inv}, \mathrm{neg}, \mathrm{sqr}, \mathrm{invpow}_n\}$ with $\mathrm{invpow}_n(x) = x^{-n}$, $n \geq 1$ and $\mathrm{sqr}(x) = x^2$. The results are shown on figures 6 and 8. For those interatomic potentials with a well-defined mathematical shape, a simple personal computer has been used and was sufficient for the computations. The right results are found in few hundred iterations on the same personal computer as described in the previous section, which means less that 2 mns with two processors and for a fitness function value less than 10^{-5}. Of course, in the case of real input-output data, the SR*v*GP will find some functions whose mathematical form is not close to well-know potential, but the SR*v*GP is sure to find a close function in the sense of uniform convergence which can be used to compute forces and other important physical quantities.

4.2. ab initio inter-atomic potentials

Let's now give an example with *ab initio* data. For example the potential energy of the Ar_2 molecule computed with the *SIESTA* code [12] in the framework of DFT (Kohn-Sham scheme with LDA approximation). We have used for SR*v*GP the same function set and the same parameters as in the previous section excepted the maximal depth which has been set to 7, and we have normalized the input-output data. The result of SR*v*GP for normalized data is shown on figure 11. This corresponds after renormalization to the following solution:

$$V(r) = 1906.22 \cdot [\frac{r}{5.71} \cdot (1 + e^{0.171021878791222}) + 0.885215174960045]^{-12} - 1141.370159$$

The fitness which is the norm of difference between the two curves is less than $6 \cdot 10^{-4}$ for few thousand iterations, which means less than 5 minutes on the same computer configuration as before. One can remark that SR*v*GP finds a power 12 which describes usually a short range repulsive interaction.

5. Conclusion

We have shown in this article that the Symbolic Regression *via* Genetic Programming (SR*v*GP) is a promising and efficient way to find a mathematical expression of the potential energy of a material system. This can be used of course for other physical quantities. Because *ab initio* calculations (Hartree-Fock, DFT, ...) are based on first principles of quantum physics, it is useful to use the results of those calculations as input-output data of SR*v*GP algorithms. This guarantees that the final function is well-suited for molecular dynamics simulations and is integrating informations from the quantum level to the mesoscopic one. Some numerical examples on simple computers have been exhibited in order to show the efficiency of SR*v*GP. Using parallel computers, our scheme is able to go further in the modelisation of inter-atomic potentials or other physical quantities with a mix of experimental and *ab initio* data for larger systems such as biological molecules or solids.

Acknowledgments

The authors thank Jean-François Osselin from University of Haute-Alsace and Michel Gondran from Electricité de France (EDF) for useful and interesting discussions.

References

[1]. Kholmurodov Kh.T. (Editor) "Molecular Simulation in Material and Biological Research". Nova Science Publishers (N.Y.). ISBN: 978-1-60741-553-4 (2009). Chapter 9. A. Kenoufi, "Finding Low-Energy Configurations of Copper-Based Bulk Metallic Glasses using Minima Hopping Global Optimization Method", pp.129-146.

[2]. A. Kenoufi, J.F. Osselin, B. Durand, "System adjustments for targeted performances combining symbolic regression and set inversion, in "Inverse problems for science and engineering", 2012.

[3]. M. Gondran, A. Kenoufi, "Numerical calculations of Holder exponents for the Weierstrass functions with (min,+)-wavelets", Trends in Applied and Computational Mathematics, 2014.

[4]. J. R. Koza and R. Poli, "Genetic Programming", in Search methodologies: Introductory tutorials in optimization and decision support techniques, page 127, Springer, 2005.

[5]. W. B. Langdon and R. Poli, "Foundations of Genetic Programming", Springer-Verlag, 2002.

[6]. Jason Brownlee, "Clever Algorithms: Nature-Inspired Programming Recipes", ISBN: 978-1-4467-8506-5.

[7]. The SR*v*GP code has been developed by the authors and Dr Jean-François Osselin from University of Haute-Alsace.

[8]. The MPI library for *ruby* language has been developed by Dr Jean-François Osselin from University of Haute-Alsace.

[9]. William Gropp, Ewing Lusk and Anthony Skjellum, "Using MPI, Portable Parallel Programming with the Message Passing Interface", 2nd edition, MIT Press, 1999.

[10]. http://www.ruby-lang.org/en/

[11]. http://www.garshol.priv.no/download/text/bnf.html

[12]. http://departments.icmab.es/leem/siesta/