Submitted to *Symmetry*. Pages 1 - 6.

OPEN ACCESS

Article

Using Symmetries (Beyond Geometric Symmetries) in Chemical Computations: Computing Parameters of Multiple Binding Sites

Andres Ortiz[1,2] **and Vladik Kreinovich**[3,⋆]

[1]Department of Mathematical Sciences, [2]Physics Department, [3]Department of Computer Science, University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA

***** Author to whom correspondence should be addressed; vladik@utep.edu, phone +1-915-747-6951, fax +1-915-747-5030.

Version May 8, 2013 submitted to Symmetry. *Typeset by LaTeX using class file mdpi.cls*

Abstract: We show how group-theoretic ideas can be naturally used to generate efficient algorithms for scientific computations. The general group-theoretic approach is illustrated on the example of determining, from the experimental data, the dissociation constants related to multiple binding sites. We also explain how the general group-theoretic approach is related to the standard (backpropagation) neural networks; this relation justifies the potential universal applicability of the group-theoretic approach.

Keywords: symmetries; group-theoretic approach; multiple binding sites; neural networks

Classification: PACS Group theory: atomic and molecular physics, 31.15.xh; mathematics, 02.20.-a

1. Why Use Group Theory in General Scientific Computations?

Use of symmetries in chemistry: a brief reminder. In many practical situations, physical systems have *symmetries*, i.e., transformations that preserve certain properties of the corresponding physical system. For example, a benzene molecule C_6H_6 does not change if we rotate it 60°: this rotation simply replaces one carbon atom by another one. The knowledge of such geometric symmetries helps in chemical computations; see, e.g., [2,3,5].

Group theory: a mathematical tool for studying symmetries. Since symmetries are useful, once we know one symmetry, it is desirable to know all the symmetries of a given physical system. In other words, once we list the properties which are preserved under the original symmetry transformation, it is desirable to find *all* the transformations that preserve these properties.

If a transformation f preserves the given properties, and the transformation g preserves these properties, then their composition $h(x) = f(g(x))$ also preserves these properties. For example, if the lowest energy level of the molecule does not change when we rotate it 60 degrees, and does not change when we rotate it 120 degrees around the same axis, then it also will not change if we first rotate it 60 degrees and then 120 degrees, to the total of 180 degrees.

Similarly, if a transformation f does not change the given properties, then the inverse transformation f^{-1} also does not change these properties. So, the set of all transformations that preserve given properties is closed under composition and inverse; such a set is called a *transformation group* or *symmetry group*. Mathematical analysis of such transformation is an important part of *group theory.*

Problems of scientific computations: a brief reminder. In this paper, we argue that group theory can be used in scientific computations beyond geometric symmetries. To explain our idea, let us briefly recall the need for scientific computations.

One of the main objectives of science is to be able to predict future behavior of physical systems. To be able to make these predictions, we must find all possible dependencies $y = F(x_1, \ldots, x_n)$ between different physical quantities. Often, we only know the general form of the dependence, i.e., we know that $y = G(x_1, \ldots, x_n, c_1, \ldots, c_m)$ for a known expression $G(x_1, \ldots, c_m)$, but we do not know the exact values of the corresponding parameters $c_1, \ldots, c_m$. These values must be determined from the empirical data. For example, Newton's equations provide a general description of how the acceleration of each celestial body depends on its spatial location, but this description contains masses c_i of celestial bodies; these masses must be determined based on the astronomical observations.

In general, to be able to predict the value of a desired quantity y for which we know the form of the dependence $y = G(x_1, \ldots, x_n, c_1, \ldots, c_m)$, we must do the following:

- first, we use the know observation $x_i^{(k)}$ and $y^{(k)}$ of x_i and y to find the parameters c_i of the corresponding dependence from the condition that $y^{(k)} \approx G(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m)$;
- after that, we measure the current values x_i of the corresponding quantities, and use these measured values and the reconstructed values of the parameters c_i to estimate y as $y = G(x_1, \ldots, x_n, c_1, \ldots, c_m)$.

In scientific computation, the first problem is known as the *inverse* problem and the second problem as the *forward* problem. Usually:

- the forward problem is reasonably straightforward: it consists of applying a previously known algorithm, while
- an inverse problem is much more complex since it requires that we solve a system of equations, and for this solution, no specific algorithm is given.

Inverse problem as the problem of finding the inverse transformation. In the idealized case, when we can ignore the measurement uncertainty, the generic inverse problem can be reformulated as follows:

- we have a transformation f corresponding to the forward problem, a transformation which maps the tuple $c = (c_1, \ldots, c_m)$ of parameters into predicted values $y_{\rm pred} = f(c)$, where each component $y_{\rm pred}^{(k)}$ has the form $y_{\rm pred}^{(k)} = G(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m)$;
- we want to find the inverse transformation f^{-1} which, based on the observed values $y^{(k)}$, computes the corresponding parameters $c_1, \ldots, c_m$.

Often, computations can be simplified if we represent the to-be-inverted transformation f as a composition. In many practical situations, we can make computations easier if, instead of directly solving a complex inverse problem, we represent it as a sequence of easier-to-solve problems.

For example, everyone knows how to solve a quadratic equation $a \cdot x^2 + b \cdot x + c = 0$. This knowledge can be effectively used if we need to solve a more complex equation $a \cdot x^4 + b \cdot x^2 + c = 0$. For that, we represent $a \cdot x^4 + b \cdot x^2 + c$ as $a \cdot y^2 + b \cdot y + c$, where $y = x^2$. Then:

- first, we solve the equation $a \cdot y^2 + b \cdot y + c$ and find y;
- next, we solve an equation $x^2 = y$ with this y and find the desired value x.

In general, if we represent a transformation f as a composition $f = f_1 \circ \ldots \circ f_n$ of transformations f_i, then the inverse transformation f^{-1} can be represented as $f_n^{-1} \circ \ldots \circ f_1^{-1}$. Thus, if we can represent the original difficult-to-invert transformation f as a composition of two easier-to-invert transformations f_i, this will simply the inversion of f.

Conclusion: transformation groups naturally appear in scientific computations. In transformation terms, solving an inverse problem means finding the inverse transformation, and simplification of this process means using compositions – and a possibility to invert each of the composed transformations. Thus, the corresponding class of transformations should be closed under composition and inverse, i.e., it should form a *transformation group*.

How group theory can help in scientific computations: general idea summarized. The inverse problem of scientific computations – the problem of estimating the parameters of the model which are the best fit for the data – is often computationally difficult to solve. From the mathematical viewpoint, this problem means finding the inverse f^{-1} to a given transformation. The computation of this inverse can be simplified if we represent f as a composition of easier-to-invert transformations $f = f_1 \circ \ldots \circ f_n$; then, we can compute f^{-1} as $f^{-1} = f_n^{-1} \circ \ldots \circ f_1^{-1}$.

2. How To Use Group Theory in General Scientific Computations: General Idea

Main idea: reminder. An inverse problem of interval computations consists of finding an inverse f^{-1} to a given transformation f. This inverse is sometimes difficult to compute. To simplify computation of f^{-1}, we try to represent f as a composition of easier-to-invert transformations f_i.

Which transformations are the easiest-to-invert. Which transformations are easier to invert? Inverting a transformation $f : \mathbb{R}^m \to \mathbb{R}^m$ means solving a system of m equations $f^{(k)}(c_1, \ldots, c_m) = y^{(k)}$ with m unknowns $c_1, \ldots, c_m$.

The simplest case is when we have a system of linear equations. In this case, there are well-known feasible algorithms for solving this system (i.e., for inverting the corresponding linear transformation). It would be nice if we could always only use linear transformations, but alas, a composition of linear transformations is always linear. So, to represent general non-linear transformations, we need to also consider some systems of non-linear equations.

For nonlinear systems, in general, the fewer unknown we have, the easier it is to solve the system. Thus, the easiest-to-solve system of non-linear equations is the system consisting of a single nonlinear equation with one unknown.

Resulting approach to scientific computing. We would like to represent an arbitrary transformation f as a composition of linear transformations and functions of one variable.

The corresponding representation is always possible. The possibility to represent an arbitrary transformation (with any given accuracy) as a composition of linear transformations and functions of one variable follows from the known fact that the standard 3-layer neural networks are universal approximators; see, e.g., [1,4]. Specifically, in a 3-layer neural network with K hidden neurons:

- we first compute K linear combinations of the inputs $y_k = \sum_{i=1}^{m} w_{ki} \cdot c_i - w_{i0}$;
- then, we apply, to each value y_k, a function $s_0(y)$ of one variable $s_0(y)$, resulting in $z_k = s_0(y_k)$; usually, a sigmoid function $s_0(y) = \dfrac{1}{1 + \exp(-y)}$ is used;
- finally, we compute a linear combination $y = \sum_{k=1}^{K} W_k \cdot z_k - W_0$.

3. Case Study: Finding Multiple Binding Sites

Case study: description. Let us show how the above general approach can applied to a specific important problem of finding multiple binding sites.

When there is a single binding site at which a ligand L can bind to a receptor R, the corresponding chemical kinetic equations $\mathrm{L} + \mathrm{R} \to \mathrm{LR}$ and $\mathrm{LR} \to \mathrm{L} + \mathrm{R}$ with intensities k^+ and k^- leads to the following equilibrium equation for the corresponding concentrations [L], [R], and [LR]: $k^+ \cdot [\mathrm{L}] \cdot [\mathrm{R}] = k^- \cdot [\mathrm{LR}]$. From this, we get $\dfrac{[\mathrm{R}]}{[\mathrm{LR}]} = \dfrac{k_d}{[\mathrm{L}]}$, where we denoted $k_d \stackrel{\text{def}}{=} \dfrac{k^-}{k^+}$. Thus, $\dfrac{[\mathrm{R}] + [\mathrm{LR}]}{[\mathrm{LR}]} = 1 + \dfrac{k_d}{[\mathrm{L}]} = \dfrac{k_d + [\mathrm{L}]}{[\mathrm{L}]}$. Thus, the bound proportion of the receptor $B \stackrel{\text{def}}{=} \dfrac{[\mathrm{LR}]}{[\mathrm{R}] + [\mathrm{LR}]}$ depends on the concentration [L] of the ligand as $B = \dfrac{[\mathrm{L}]}{k_d + [\mathrm{L}]}$.

For the case of several (S) binding sites, B is a linear combination of terms corresponding to different binding sites, i.e.,

$$B = \sum_{s=1}^{S} \frac{R_s \cdot [\mathrm{L}]}{k_{ds} + [\mathrm{L}]} \tag{1}$$

for appropriate values R_s and k_{ds}.

Inverse problem corresponding to the case study. The problem is to find the values R_s and k_{ds} from the observations. In other words, we observe the bound proportions $y^{(k)}$ for different ligand concentrations $[\mathrm{L}] = x^{(k)}$, and we want to find the values R_s and k_{ds} for which

$$y^{(k)} = \sum_{s=1}^{S} \frac{R_s \cdot x^{(k)}}{k_{ds} + x^{(k)}}. \tag{2}$$

How to use group-theoretic ideas to simplify the corresponding computations: analysis of the problem. The system (2) is a difficult-to-solve system of nonlinear equations with $2S$ unknowns. To simplifying the solution of this system, let us represent its solution as a composition of linear transformations and functions of one variable.

By adding all S fractions $\frac{R_s \cdot x}{k_{ds} + x}$, we get a ratio of two polynomials $\frac{P(x)}{Q(x)}$. Here, $Q(x)$ is the product of all S denominators $x + k_{ds}$, and is, thus, a S-th order polynomial with the leading term x^S:

$$Q(x) = x^S + q_{S-1} \cdot x^{S-1} + \ldots + q_1 \cdot x + q_0. \tag{3}$$

Similarly, since $P(x)$ is divisible by x, we get $P(x) = p_S \cdot x^S + p_{S-1} \cdot x^{S-1} + \ldots + p_1 \cdot x$.

The equations $y^{(k)} = \frac{P(x^{(k)})}{Q(x^{(k)})}$ can be equivalently represented as $y^{(k)} \cdot Q(x^{(k)}) = P(x^{(k)})$, i.e., as

$$y^{(k)} \cdot (x^{(k)})^S + q_{S-1} \cdot y^{(k)} \cdot (x^{(k)})^{S-1} + \ldots + q_1 \cdot y^{(k)} \cdot x^{(k)} + q_0 \cdot y^{(k)} =$$

$$p_S \cdot (x^{(k)})^S + p_{S-1} \cdot (x^{(k)})^{S-1} + \ldots + p_1 \cdot x^{(k)}. \tag{4}$$

This is a system of linear equations with $2S$ unknowns p_i and q_i. Solving this system of linear equations is relatively easy.

Once we solve this linear system and find the values q_i, we can find the parameters k_{ds} from the condition that for $x = -k_{ds}$, we have $x + k_{ds} = 0$ and thus, the product $Q(x)$ of all such terms is equal to 0. The equation $Q(-k_{ds}) = 0$ is a nonlinear equation with one unknown, i.e., exactly the type of nonlinear equation that we want to solve.

Finally, once we find all the values k_{ds}, the equation (2) becomes a linear system of equations for the remaining unknowns R_s.

Thus, the decomposition of the original difficult-to-invert transformation into a composition of easier-to-invert transformations (linear transformations and functions of one variable) leads to the following algorithm for computing the parameters of multiple binding sites.

Inverse problem corresponding to the case study: resulting algorithm. We start with the values $y^{(k)}$ of the bound proportion corresponding to different ligand concentrations $x^{(k)}$. Our objective is to find the parameters R_s and k_{ds} of different binding sites $s = 1, \ldots, S$. To compute these parameters, we do the following:

- first, we solve the linear system (4) with $2S$ unknowns p_i and q_i;
- we then use the computed values q_i to form the polynomial (3) and to solve the equation $Q(-x) = 0$ with one unknown x; as a result, we get $2S$ solutions k_{ds};
- we then substitute the resulting values k_{ds} into the formula (1) and solve the resulting system of S linear equations with S unknowns R_s.

Comment. Our numerical experiments confirmed the computational efficiency of the new algorithm.

Acknowledgements

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research. The authors are thankful to Mahesh Narayan for his help and to Larry Ellzey and Ming-Ying Leung for their encouragement.

References

1. Bishop, C.M. *Pattern Recognition and Machine Learning*, Springer: New York, New York, USA, 2006.
2. Jaffé, H.H.; MacKenzie, R.E. *Symmetry in Chemistry*, Dover: New York, New York, USA, 2012.
3. Kettle, S.F.A. *Symmetry and Structure: Readable Group Theory for Chemists*, Wiley: New York, New York, USA, 2007.
4. Nguyen, H.T.; Kreinovich, V. *Applications of continuous mathematics to computer science*, Kluwer: Dordrecht, Netherlands, 1997.
5. Wigner, E.P. *Group Theory and Its Application to the Quantum Mechanics of Atomic Spectra*; Academic Press: Waltham: Massachusetts, USA, 1959.