

SANDIA REPORT

SAND2021-9473

Printed August, 2021



Sandia
National
Laboratories

AEVmod – Atomic Environment Vector Module Documentation

Habib N. Najm, Yoona Yang

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: <http://www.osti.gov/scitech>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: <https://classic.ntis.gov/help/order-methods>



ABSTRACT

This report outlines the mathematical formulation for the atomic environment vector (AEV) construction used in the `aevmol` software package. The AEV provides a summary of the geometry of a molecule or atomic configuration. We also present the formulation for the analytical Jacobian of the AEV with respect to the atomic Cartesian coordinates. The software provides functionality for both the AEV and AEV-Jacobian, as well as the AEV-Hessian which is available via reliance on the third party library `Sacado`.

ACKNOWLEDGMENT

This work is supported as part of the Computational Chemical Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

CONTENTS

1. Introduction	6
2. Mathematical Formulation	7
2.1. Atomic Environment Vector.....	7
2.2. Analytical Jacobian.....	10
2.3. A simple AEV example	14
3. Code Use Example	16
References	19

1. INTRODUCTION

The `aevmod` package provides functionality for the representation of geometrical structure of a molecule, or an atomic configuration, using the atomic environment vector (AEV) construction of Smith *et al.* [6]. It also provides functionality for computation of the AEV Jacobian and Hessian with respect to Cartesian atomic coordinates. This report outlines the AEV construction, with detailed formulation of the AEV representation as well as its analytical Jacobian. The analytical Hessian formulation is beyond the scope of this report, and is substituted here with reliance on automatic differentiation of the underlying C++ AEV code using Sacado [1] to arrive at the Hessian. The AEV formulation, following [6], is based on the Behler and Parinello [2–5] symmetry functions (SFs), which ensure automatic satisfaction of translational and rotational invariances in the AEV construction. This formulation, including both the AEV and its Jacobian, are detailed in § 2, while highlights of the code structure and an illustrative example implementation are discussed in § 3.

2. MATHEMATICAL FORMULATION

We begin with a description of the formulation of the AEV, followed by that of the analytical Jacobian, and end with a simple example.

2.1. Atomic Environment Vector

Consider a configuration of N atoms $\mathcal{C} = (A_1, \dots, A_N)$ in some geometrical arrangement. This is an ordered vector, where e.g. for $\mathcal{C} = (\text{H}, \text{H}, \text{O})$, we have $A_1 = A_2 = \text{H}$, and $A_3 = \text{O}$. It can be a molecule, e.g. H_2O , or a combination of a molecule and an atom, e.g. OH and H , or H_2 and O ; or simply three unbound atoms H , H , and O . In general, let the location of atom A_i be the vector $x_i \in \mathbb{R}^3$. The system is fully defined by $(\mathcal{C}, \mathbf{x})$ where $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^{3N}$, is the vector of coordinates of all N atoms in \mathcal{C} . Further, we define the set of atom types in the system as $T = \{T_1, \dots, T_n\}$, and the number of atoms of each type $\tau \in T$ as N_τ . For present purposes, we define the type of any atom A_i as

$$\mathcal{T}_i := \mathcal{T}(A_i) \equiv A_i, \quad (2.1.1)$$

thus, again, for the above configuration $(\text{H}, \text{H}, \text{O})$, we have the set of atom types $T := \{\text{O}, \text{H}\}$. Next, let the index set of atoms of type τ in the configuration \mathcal{C} be S_τ . Thus for, $\mathcal{C} = (A_1, \dots, A_N)$, and $T = \{T_1, \dots, T_n\}$,

$$S_\tau := \{i \mid \mathcal{T}_i = \tau\}, \quad \tau = T_1, \dots, T_n; \quad i = 1, \dots, N \quad (2.1.2)$$

Similarly, we define the index set of pair-wise atom indices, with $\tau = T_1, \dots, T_n$; $\kappa = \tau, \dots, T_n$; as follows

$$S_{\tau, \kappa} := \{(j, k) \mid \mathcal{T}_j = \tau, \mathcal{T}_k = \kappa, j = 1, \dots, N-1; k = j+1, \dots, N\} \quad (2.1.3)$$

For example, for a system with $T = \{\text{C}, \text{H}, \text{O}\}$, and the two configurations $(\text{H}, \text{H}, \text{O})$ and $(\text{C}, \text{O}, \text{O})$, we have the following index sets:

\mathcal{C}	S_{C}	S_{H}	S_{O}	$S_{\text{C,C}}$	$S_{\text{C,H}}$	$S_{\text{C,O}}$	$S_{\text{H,H}}$	$S_{\text{H,O}}$	$S_{\text{O,O}}$
$(\text{H}, \text{H}, \text{O})$	$\{\}$	$\{1, 2\}$	$\{3\}$	$\{\}$	$\{\}$	$\{\}$	$\{(1, 2)\}$	$\{(1, 3), (2, 3)\}$	$\{\}$
$(\text{C}, \text{O}, \text{O})$	$\{1\}$	$\{\}$	$\{2, 3\}$	$\{\}$	$\{\}$	$\{(1, 2), (1, 3)\}$	$\{\}$	$\{\}$	$\{(2, 3)\}$

Similarly, for $T = \{\text{C}, \text{H}, \text{O}, \text{N}\}$, and $\mathcal{C} = (\text{H}, \text{C}, \text{O}, \text{O}, \text{H})$, e.g. this can be the molecule formic acid, we have

\mathcal{C}	S_{C}	S_{H}	S_{O}	S_{N}
$(\text{H}, \text{C}, \text{O}, \text{O}, \text{H})$	$\{2\}$	$\{1, 5\}$	$\{3, 4\}$	$\{\}$

\mathcal{C}	$S_{C,C}$	$S_{C,H}$	$S_{C,O}$	$S_{C,N}$	$S_{H,H}$
(H,C,O,O,H)	{}	{(1,2),(2,5)}	{(2,3),(2,4)}	{}	{(1,5)}

\mathcal{C}	$S_{H,O}$	$S_{H,N}$	$S_{O,O}$	$S_{O,N}$	$S_{N,N}$
(H,C,O,O,H)	{(1,3),(1,4),(3,5),(4,5)}	{}	{(3,4)}	{}	{}

Note that, with n atom types, there are n index sets S_τ , and $m = n(n+1)/2$ index sets $S_{\tau,\kappa}$.

We define the AEV of atom A_i in a configuration as $y_i(\mathcal{C}, \mathbf{x}) \in \mathbb{R}^M$. The AEV has both radial and angular components, and is constructed as follows. Consider any atom A_i . Let us write the radial components of y_i for pairings with each available atom type separately, according to the S_τ index sets. Similarly, we write angular components of y_i , for groupings with pairs of atom types separately according to the index sets $S_{\tau,\kappa}$. In each case this is done by summing contributions of the pairings/groupings within each set.

First, define the cutoff function

$$f_c(R_{ij}, R_c) = \begin{cases} 0.5 \cos\left(\frac{\pi R_{ij}}{R_c}\right) + 0.5 & \text{for } R_{ij} \leq R_c \\ 0.0 & \text{for } R_{ij} > R_c \end{cases} \quad (2.1.4)$$

where $R_{ij} = |x_i - x_j|$ is the Euclidian distance between the two atoms (A_i, A_j) , $x_i \in \mathbb{R}^3$ is the location of atom A_i , similarly for x_j of A_j , and R_c is a cutoff radius.

Then with the parameter vector $\mu = (\eta^R, \rho^R) \in \mathbb{R}^2$, to be defined below, define radial components of y_i for pairings with atoms of type $\tau \in T$ as

$$y_i^{R,\tau,\mu} = \sum_{j \in S_\tau, j \neq i} e^{-\eta^R(R_{ij}-\rho^R)^2} f_c(R_{ij}, R_c^R) \quad (2.1.5)$$

with $\mu \in \{\mu_1, \dots, \mu_{M_R}\}$ in general, and where R_c^R is a radial-SF specific cutoff radius. Further, with a second parameter vector $\nu = (\eta^A, \rho^A, \zeta, \alpha) \in \mathbb{R}^4$, we define angular components of y_i as

$$y_i^{A,\tau,\kappa,\nu} = \sum_{\substack{(j,k) \in S_{\tau,\kappa} \\ j \neq i, k \neq i}} (0.5 + 0.5 \cos(\theta_{ijk} - \alpha))^\zeta e^{-\eta^A(0.5(R_{ij}+R_{ik})-\rho^A)^2} f_c(R_{ij}, R_c^A) f_c(R_{ik}, R_c^A) \quad (2.1.6)$$

with $\nu \in \{\nu_1, \dots, \nu_{M_A}\}$, and R_c^A is an angular-SF specific cutoff radius. Note that θ_{ijk} is the angle, centered on A_i , between the two vectors $x_{ij} := x_j - x_i$ and $x_{ik} := x_k - x_i$, thus

$$\theta_{ijk} := \arccos \frac{x_{ij} \cdot x_{ik}}{|x_{ij}| |x_{ik}|}, \quad \text{such that } \theta_{ijk} \in [0, \pi]. \quad (2.1.7)$$

Note that this angular SF formulation differs from that in Smith *et al.* [6] in two ways that ultimately do not matter. First, we have dispensed with the extra factor of 2 in front of the summation. In their formulation, the factor in front of the sum is $2^{1-\zeta}$, which is $2/2^\zeta$. The denominator is absorbed in the formulation of the summand in the above angular SF expression, which leaves a constant factor of 2 in front of the sum. A scaling factor like that among all the angular SF components is

not particularly useful, as, when this feature vector is used as input to a neural network (NN), the NN training would make up for it anyway with appropriate scaling if absent. The other difference is that their summation includes terms like (j, k) and (k, j) , which are excluded above by the definition of $S_{\tau, \kappa}$, where we have $k = j + 1, \dots, N$ by construction, thus enforcing that the second index is always greater than the first in any element of $S_{\tau, \kappa}$. This does not imply any meaningful change, however, since the symmetry of the summand with respect to commutation of the indices is clear, so that the summand is identical for any two pairs (j, k) and (k, j) . Effectively this translates to another doubling relative to the above expression, which is again inconsequential. Thus, the above expression is simpler, dispensing with the factor in front of the sum, and with the logic for avoidance of repeated computation of identical terms.

The full length of the AEV is

$$M = M_R n + M_A m \quad (2.1.8)$$

and, with

$$y_i^{R, \tau} := (y_i^{R, \tau, \mu_1}, \dots, y_i^{R, \tau, \mu_{M_R}}) \in \mathbb{R}^{M_R} \quad (2.1.9)$$

$$y_i^{A, \tau, \kappa} := (y_i^{A, \tau, \kappa, v_1}, \dots, y_i^{A, \tau, \kappa, v_{M_A}}) \in \mathbb{R}^{M_A} \quad (2.1.10)$$

we can then write the full AEV as

$$y_i = (y_i^{R, \tau_1}, \dots, y_i^{R, \tau_n}, y_i^{A, (\tau, \kappa)_1}, \dots, y_i^{A, (\tau, \kappa)_m}) \in \mathbb{R}^M. \quad (2.1.11)$$

Thus, for example, for $T = \{\text{C, H, O, N}\}$, hence $n = 4$, if we use, as in the supplementary materials in [6], $M_R = 8$ (in their notation: a fixed η and 8 different R_s values) and $M_A = 8$, (in their notation: fixed ζ, η, R_s and 8 different θ_s values), we have $M = 112$, for any molecule/configuration comprised of the elements of T . Similarly, as used in the main text in [6], p. 3200, with $M_R = 32$ (their notation: fixed η and 32 R_s), and $M_A = 64$ (their notation: fixed η, ζ ; 8 R_s values, and 8 θ_s values, giving $8 \times 8 = 64$) we have $M = 768$.

Note that, with h being the spacing of the symmetry functions, choosing $\delta/h \approx 1$ defines the degree of overlap among them. We find $\delta/h = 2/3$ reasonable, being also close to the choice in [6]. Similarly, ζ provides a strength of decay of the cosine function. Increasing ζ gives a steeper drop in the function, and thus less overlap between neighboring angular SFs. Reasonable nominal values of the AEV parameters are

- $R_c^R = 4.6, R_c^A = 3.1$
- $\zeta = 8$
- $n_{\rho^R} = 32, h = R_c^R/n_{\rho^R}, \eta^R = 1.0/\delta^2, \delta = 2h/3, \rho_i^R = (i - \frac{1}{2})h, i = 1, \dots, n_{\rho^R}$
- $n_{\rho^A} = 8, h = R_c^A/n_{\rho^A}, \eta^A = 1.0/\delta^2, \delta = 2h/3, \rho_i^A = (i - \frac{1}{2})h, i = 1, \dots, n_{\rho^A}$
- $n_\alpha = 8, h = \pi/(n_\alpha - 1), \alpha_i = (i - 1)h, i = 1, \dots, n_\alpha$

2.2. Analytical Jacobian

Let us next write down the analytical Jacobian for the above AEV construction. We have so far expressed the 3D coordinates of an atom ℓ as x_ℓ . Here will expressly derive the formulation in terms of the 3 spatial coordinates, where we define $x := (u, v, w)$. Considering a configuration with N atoms, our key requirement is to write-out the gradients

$$g_{ir\ell} := \left(\frac{\partial y_{ir}}{\partial u_\ell}, \frac{\partial y_{ir}}{\partial v_\ell}, \frac{\partial y_{ir}}{\partial w_\ell} \right)^T, \quad i, \ell = 1, \dots, N; \quad r = 1, \dots, M \quad (2.2.1)$$

where with $\mu = (\eta^R, \rho^R)$ and $\nu = (\eta^A, \rho^A, \zeta, \alpha)$, and for AEV designed values given by $\mu \in \{\mu_1, \dots, \mu_{M_R}\}$ and $\nu \in \{\nu_1, \dots, \nu_{M_A}\}$, and with $\sigma_\ell \in \{u_\ell, v_\ell, w_\ell\}$ we have

$$\frac{\partial y_{ir}}{\partial \sigma_\ell} = \begin{cases} \frac{\partial}{\partial \sigma_\ell} y_i^{R, \tau_p, \mu_q} & p = 1, \dots, n; q = 1, \dots, M_R; r = (p-1)M_R + q = 1, \dots, M_R n \\ \frac{\partial}{\partial \sigma_\ell} y_i^{A, (\tau, \kappa)_p \nu_q} & p = 1, \dots, m; q = 1, \dots, M_A; r = M_R n + (p-1)M_A + q = M_R n + 1, \dots, M \end{cases}$$

Further, with

$$\chi_{ijq}^R := e^{-\eta_q^R (R_{ij} - \rho_q^R)^2} f_c(R_{ij}, R_c^R) \quad (2.2.2)$$

we have

$$\frac{\partial}{\partial \sigma_\ell} y_i^{R, \tau_p, \mu_q} = \sum_{j \in S_{\tau_p}, j \neq i} \frac{\partial}{\partial \sigma_\ell} \chi_{ijq}^R \quad (2.2.3)$$

and, with

$$\chi_{ijkq}^A := (0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q e^{-\eta_q^A (0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} f_c(R_{ij}, R_c^A) f_c(R_{ik}, R_c^A) \quad (2.2.4)$$

we have

$$\frac{\partial}{\partial \sigma_\ell} y_i^{A, (\tau, \kappa)_p \nu_q} = \sum_{\substack{(j,k) \in S_{(\tau, \kappa)_p} \\ j \neq i, k \neq i}} \frac{\partial}{\partial \sigma_\ell} \chi_{ijkq}^A. \quad (2.2.5)$$

Moreover, we have

$$\frac{\partial}{\partial \sigma_\ell} \chi_{ijq}^R = \begin{cases} \frac{\partial}{\partial \sigma_i} \chi_{ijq}^R & \text{for } \ell = i \\ \frac{\partial}{\partial \sigma_j} \chi_{ijq}^R & \text{for } \ell = j \\ 0 & \text{otherwise} \end{cases} \quad (2.2.6)$$

and

$$\frac{\partial}{\partial \sigma_\ell} \chi_{ijkq}^A = \begin{cases} \frac{\partial}{\partial \sigma_i} \chi_{ijkq}^A & \text{for } \ell = i \\ \frac{\partial}{\partial \sigma_j} \chi_{ijkq}^A & \text{for } \ell = j \\ \frac{\partial}{\partial \sigma_k} \chi_{ijkq}^A & \text{for } \ell = k \\ 0 & \text{otherwise.} \end{cases} \quad (2.2.7)$$

Thus, we need to derive $\nabla_{x_i} \chi_{ijq}^R$, $\nabla_{x_j} \chi_{ijq}^R$, $\nabla_{x_i} \chi_{ijkq}^A$, $\nabla_{x_j} \chi_{ijkq}^A$, and $\nabla_{x_k} \chi_{ijkq}^A$.

We have, to begin with,

$$\begin{aligned} \frac{\partial}{\partial \sigma_i} \chi_{ijq}^R &= \frac{\partial}{\partial \sigma_i} \left(e^{-\eta_q^R (R_{ij} - \rho_q^R)^2} f_c(R_{ij}, R_c^R) \right) \\ &= \frac{\partial}{\partial \sigma_i} \left(e^{-\eta_q^R (R_{ij} - \rho_q^R)^2} \right) f_c(R_{ij}, R_c^R) + e^{-\eta_q^R (R_{ij} - \rho_q^R)^2} \frac{\partial}{\partial \sigma_i} \left(f_c(R_{ij}, R_c^R) \right) \\ &= -2\eta_q^R (R_{ij} - \rho_q^R) \frac{\partial R_{ij}}{\partial \sigma_i} e^{-\eta_q^R (R_{ij} - \rho_q^R)^2} f_c(R_{ij}, R_c^R) + e^{-\eta_q^R (R_{ij} - \rho_q^R)^2} \frac{\partial}{\partial \sigma_i} \left(f_c(R_{ij}, R_c^R) \right) \end{aligned}$$

and

$$\frac{\partial}{\partial \sigma_i} R_{ij} = \frac{\partial}{\partial \sigma_i} |x_i - x_j| = \frac{\partial}{\partial \sigma_i} \left((u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2 \right)^{1/2} \quad (2.2.8)$$

$$= \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial \sigma_i} (u_i - u_j)^2 + \frac{\partial}{\partial \sigma_i} (v_i - v_j)^2 + \frac{\partial}{\partial \sigma_i} (w_i - w_j)^2 \right) \quad (2.2.9)$$

or

$$\frac{\partial}{\partial u_i} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial u_i} (u_i - u_j)^2 \right) = \frac{u_i - u_j}{R_{ij}} \quad (2.2.10)$$

$$\frac{\partial}{\partial v_i} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial v_i} (v_i - v_j)^2 \right) = \frac{v_i - v_j}{R_{ij}} \quad (2.2.11)$$

$$\frac{\partial}{\partial w_i} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial w_i} (w_i - w_j)^2 \right) = \frac{w_i - w_j}{R_{ij}} \quad (2.2.12)$$

Further, for notational convenience, define the (modified) Heaviside function, for $x \in \mathbb{R}$

$$H(x) := \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases} \quad (2.2.13)$$

such that

$$f_c(R_{ij}, R_c) = \left[0.5 \cos \left(\frac{\pi R_{ij}}{R_c} \right) + 0.5 \right] H(R_c - R_{ij}) \quad (2.2.14)$$

and

$$\frac{\partial}{\partial \sigma_i} \left(f_c(R_{ij}, R_c^R) \right) = -0.5 \frac{\pi}{R_c^R} \frac{\partial R_{ij}}{\partial \sigma_i} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.15)$$

so that

$$\frac{\partial}{\partial u_i} \left(f_c(R_{ij}, R_c^R) \right) = -0.5 \frac{\pi}{R_c^R} \frac{u_i - u_j}{R_{ij}} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.16)$$

$$\frac{\partial}{\partial v_i} \left(f_c(R_{ij}, R_c^R) \right) = -0.5 \frac{\pi}{R_c^R} \frac{v_i - v_j}{R_{ij}} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.17)$$

$$\frac{\partial}{\partial w_i} \left(f_c(R_{ij}, R_c^R) \right) = -0.5 \frac{\pi}{R_c^R} \frac{w_i - w_j}{R_{ij}} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.18)$$

Similarly,

$$\frac{\partial}{\partial \sigma_j} \chi_{ijq}^R = -2\eta_q^R(R_{ij} - \rho_q^R) \frac{\partial R_{ij}}{\partial \sigma_j} e^{-\eta_q^R(R_{ij} - \rho_q^R)^2} f_c(R_{ij}, R_c^R) + e^{-\eta_q^R(R_{ij} - \rho_q^R)^2} \frac{\partial}{\partial \sigma_j} (f_c(R_{ij}, R_c^R))$$

and

$$\frac{\partial}{\partial \sigma_j} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial \sigma_j} (u_i - u_j)^2 + \frac{\partial}{\partial \sigma_j} (v_i - v_j)^2 + \frac{\partial}{\partial \sigma_j} (w_i - w_j)^2 \right) \quad (2.2.19)$$

or

$$\frac{\partial}{\partial u_j} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial u_j} (u_i - u_j)^2 \right) = -\frac{u_i - u_j}{R_{ij}} \quad (2.2.20)$$

$$\frac{\partial}{\partial v_j} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial v_j} (v_i - v_j)^2 \right) = -\frac{v_i - v_j}{R_{ij}} \quad (2.2.21)$$

$$\frac{\partial}{\partial w_j} R_{ij} = \frac{1}{2R_{ij}} \left(\frac{\partial}{\partial w_j} (w_i - w_j)^2 \right) = -\frac{w_i - w_j}{R_{ij}} \quad (2.2.22)$$

and

$$\frac{\partial}{\partial \sigma_j} (f_c(R_{ij}, R_c^R)) = -0.5 \frac{\pi}{R_c^R} \frac{\partial R_{ij}}{\partial \sigma_j} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.23)$$

so that

$$\frac{\partial}{\partial u_j} (f_c(R_{ij}, R_c^R)) = 0.5 \frac{\pi}{R_c^R} \frac{u_i - u_j}{R_{ij}} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.24)$$

$$\frac{\partial}{\partial v_j} (f_c(R_{ij}, R_c^R)) = 0.5 \frac{\pi}{R_c^R} \frac{v_i - v_j}{R_{ij}} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}) \quad (2.2.25)$$

$$\frac{\partial}{\partial w_j} (f_c(R_{ij}, R_c^R)) = 0.5 \frac{\pi}{R_c^R} \frac{w_i - w_j}{R_{ij}} \sin \left(\frac{\pi R_{ij}}{R_c^R} \right) H(R_c^R - R_{ij}). \quad (2.2.26)$$

Consider then

$$\begin{aligned} \frac{\partial}{\partial \sigma_\ell} \chi_{ijkq}^A &= \frac{\partial}{\partial \sigma_\ell} \left((0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q e^{-\eta_q^A(0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} f_c(R_{ij}, R_c^A) f_c(R_{ik}, R_c^A) \right) \\ &= \frac{\partial}{\partial \sigma_\ell} \left((0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q \right) e^{-\eta_q^A(0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} f_c(R_{ij}, R_c^A) f_c(R_{ik}, R_c^A) \\ &\quad + (0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q \frac{\partial}{\partial \sigma_\ell} \left(e^{-\eta_q^A(0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} \right) f_c(R_{ij}, R_c^A) f_c(R_{ik}, R_c^A) \\ &\quad + (0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q e^{-\eta_q^A(0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} \frac{\partial}{\partial \sigma_\ell} \left(f_c(R_{ij}, R_c^A) \right) f_c(R_{ik}, R_c^A) \\ &\quad + (0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q e^{-\eta_q^A(0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} f_c(R_{ij}, R_c^A) \frac{\partial}{\partial \sigma_\ell} \left(f_c(R_{ik}, R_c^A) \right). \end{aligned}$$

With the derivatives of $f_c()$ already done above, we only need to derive, for $\ell = i, j, k$,

$$\frac{\partial}{\partial \sigma_\ell} \left((0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q \right) = -\frac{1}{2} \zeta_q (0.5 + 0.5 \cos(\theta_{ijk} - \alpha_q)) \zeta_q^{-1} \sin(\theta_{ijk} - \alpha_q) \frac{\partial}{\partial \sigma_\ell} \theta_{ijk}$$

and, with $\gamma_{ijkq} := 0.5(R_{ij} + R_{ik}) - \rho_q^A$,

$$\begin{aligned}
\frac{\partial}{\partial \sigma_\ell} \left(e^{-\eta_q^A (0.5(R_{ij} + R_{ik}) - \rho_q^A)^2} \right) &= e^{-\eta_q^A \gamma_{ijkq}^2} \frac{\partial}{\partial \sigma_\ell} \left(-\eta_q^A \gamma_{ijkq}^2 \right) \\
&= -\eta_q^A e^{-\eta_q^A \gamma_{ijkq}^2} 2\gamma_{ijkq} \frac{\partial}{\partial \sigma_\ell} \gamma_{ijkq} \\
&= -\eta_q^A e^{-\eta_q^A \gamma_{ijkq}^2} \gamma_{ijkq} \frac{\partial}{\partial \sigma_\ell} (R_{ij} + R_{ik}) \\
&= -\eta_q^A e^{-\eta_q^A \gamma_{ijkq}^2} \gamma_{ijkq} \left(\frac{\partial}{\partial \sigma_\ell} R_{ij} + \frac{\partial}{\partial \sigma_\ell} R_{ik} \right). \quad (2.2.27)
\end{aligned}$$

Then, we have already $\frac{\partial}{\partial \sigma_\ell} R_{ij}$ for $\ell = i, j$ above, and it's zero for $\ell = k$. Similarly, we can write $\frac{\partial}{\partial \sigma_\ell} R_{ik}$ for $\ell = i, k$, and it's zero for $\ell = j$. This leaves only $\frac{\partial}{\partial \sigma_\ell} \theta_{ijk}$ to be derived, where, with $\theta_{ijk} \in [0, \pi]$, $x_{ij} = x_j - x_i$, $x_{ik} = x_k - x_i$,

$$\frac{\partial}{\partial \sigma_\ell} \theta_{ijk} = \frac{\partial}{\partial \sigma_\ell} \arccos \frac{x_{ij} \cdot x_{ik}}{|x_{ij}| |x_{ik}|} \quad (2.2.28)$$

and, given the identity $d(\cos^{-1} u)/dx = -(du/dx)/\sqrt{1-u^2}$, we have

$$\begin{aligned}
\frac{\partial}{\partial \sigma_\ell} \theta_{ijk} &= -\frac{\partial}{\partial \sigma_\ell} \left(\frac{x_{ij} \cdot x_{ik}}{|x_{ij}| |x_{ik}|} \right) \left[1 - \left(\frac{x_{ij} \cdot x_{ik}}{|x_{ij}| |x_{ik}|} \right)^2 \right]^{-1/2} \\
&= -\frac{|x_{ij}| |x_{ik}| \frac{\partial}{\partial \sigma_\ell} (x_{ij} \cdot x_{ik}) - (x_{ij} \cdot x_{ik}) \frac{\partial}{\partial \sigma_\ell} (|x_{ij}| |x_{ik}|)}{(|x_{ij}| |x_{ik}|)^2} \left[1 - \left(\frac{x_{ij} \cdot x_{ik}}{|x_{ij}| |x_{ik}|} \right)^2 \right]^{-1/2}
\end{aligned}$$

which, since $|x_{ij}| = |x_j - x_i| = |x_i - x_j| = R_{ij}$, and $|x_{ik}| = R_{ik}$, and $x_{ij} \cdot x_{ik} = |x_{ij}| |x_{ik}| \cos \theta_{ijk} = R_{ij} R_{ik} \cos \theta_{ijk}$, and since $1 - \cos^2 \theta_{ijk} = \sin^2 \theta_{ijk}$, becomes

$$\frac{\partial}{\partial \sigma_\ell} \theta_{ijk} = -\frac{\frac{\partial}{\partial \sigma_\ell} (x_{ij} \cdot x_{ik}) - \cos \theta_{ijk} \frac{\partial}{\partial \sigma_\ell} (R_{ij} R_{ik})}{R_{ij} R_{ik}} \frac{1}{\sin \theta_{ijk}} \quad (2.2.29)$$

with

$$\frac{\partial}{\partial \sigma_\ell} (R_{ij} R_{ik}) = R_{ik} \frac{\partial}{\partial \sigma_\ell} R_{ij} + R_{ij} \frac{\partial}{\partial \sigma_\ell} R_{ik} \quad (2.2.30)$$

which uses derivatives of R_{ij} and R_{ik} already evaluated.

This leaves only

$$\frac{\partial}{\partial \sigma_\ell} (x_{ij} \cdot x_{ik}). \quad (2.2.31)$$

We write

$$x_{ij} \cdot x_{ik} = (u_{ij}, v_{ij}, w_{ij}) \cdot (u_{ik}, v_{ik}, w_{ik}) \quad (2.2.32)$$

where, $u_{ij} := u_j - u_i$, and similarly for the others, so that

$$x_{ij} \cdot x_{ik} = u_{ij} u_{ik} + v_{ij} v_{ik} + w_{ij} w_{ik} \quad (2.2.33)$$

so that

$$\frac{\partial}{\partial \sigma_\ell}(x_{ij} \cdot x_{ik}) = \frac{\partial}{\partial \sigma_\ell}(u_{ij}u_{ik} + v_{ij}v_{ik} + w_{ij}w_{ik}) \quad (2.2.34)$$

$$= \frac{\partial}{\partial \sigma_\ell}(u_{ij}u_{ik}) + \frac{\partial}{\partial \sigma_\ell}(v_{ij}v_{ik}) + \frac{\partial}{\partial \sigma_\ell}(w_{ij}w_{ik}) \quad (2.2.35)$$

$$= u_{ik} \frac{\partial u_{ij}}{\partial \sigma_\ell} + u_{ij} \frac{\partial u_{ik}}{\partial \sigma_\ell} + v_{ik} \frac{\partial v_{ij}}{\partial \sigma_\ell} + v_{ij} \frac{\partial v_{ik}}{\partial \sigma_\ell} + w_{ik} \frac{\partial w_{ij}}{\partial \sigma_\ell} + w_{ij} \frac{\partial w_{ik}}{\partial \sigma_\ell} \quad (2.2.36)$$

all of which are trivially known. Thus

$$\frac{\partial}{\partial u_i}(x_{ij} \cdot x_{ik}) = -u_{ik} - u_{ij} \quad (2.2.37)$$

$$\frac{\partial}{\partial v_i}(x_{ij} \cdot x_{ik}) = -v_{ik} - v_{ij} \quad (2.2.38)$$

$$\frac{\partial}{\partial w_i}(x_{ij} \cdot x_{ik}) = -w_{ik} - w_{ij} \quad (2.2.39)$$

$$\frac{\partial}{\partial u_j}(x_{ij} \cdot x_{ik}) = u_{ik} \quad (2.2.40)$$

$$\frac{\partial}{\partial v_j}(x_{ij} \cdot x_{ik}) = v_{ik} \quad (2.2.41)$$

$$\frac{\partial}{\partial w_j}(x_{ij} \cdot x_{ik}) = w_{ik} \quad (2.2.42)$$

$$\frac{\partial}{\partial u_k}(x_{ij} \cdot x_{ik}) = u_{ij} \quad (2.2.43)$$

$$\frac{\partial}{\partial v_k}(x_{ij} \cdot x_{ik}) = v_{ij} \quad (2.2.44)$$

$$\frac{\partial}{\partial w_k}(x_{ij} \cdot x_{ik}) = w_{ij} \quad (2.2.45)$$

With all this, then for any given molecular configuration in a stated geometry, we can compute both the AEV and its spatial derivatives.

2.3. A simple AEV example

Consider a case with $T = \{H, O\}$, and $\mathcal{C} = (O, H, H)$, and let's work it out in detail. We begin by writing down the radial and angular index sets as follows:

\mathcal{C}	S_H	S_O	$S_{H,H}$	$S_{H,O}$	$S_{O,O}$
(O,H,H)	{2,3}	{1}	{(2,3)}	{(1,2),(1,3)}	{}

Then, consider $M_R = 2$, $M_A = 4$, so that the full set of SF parameters is:

R_c	$R_c^R = 4.6$	$R_c^A = 3.1$
$\mu = (\eta^R, \rho^R)$	$\mu_1 = (0.42533081^{-2}, 1.15)$	$\mu_2 = (0.42533081^{-2}, 3.45)$
$\nu = (\eta^A, \rho^A, \zeta, \alpha)$	$\nu_1 = (0.93652445^{-2}, 0.775, 8, 0 \text{ rad})$ $\nu_3 = (0.93652445^{-2}, 2.325, 8, 0 \text{ rad})$	$\nu_2 = (0.93652445^{-2}, 0.775, 8, \pi \text{ rad})$ $\nu_4 = (0.93652445^{-2}, 2.325, 8, \pi \text{ rad})$

and that the AEV, for each atom i in \mathcal{C} is given by:

$$y_i = (y_i^{R,H}, y_i^{R,O}, y_i^{A,H,H}, y_i^{A,H,O}, y_i^{A,O,O}) \quad (2.3.1)$$

or, fully expanded with $M_R = 2$ radial SFs and $M_A = 4$ angular SFs,

$$y_i = (y_i^{R,H,\mu_1}, y_i^{R,H,\mu_2}, y_i^{R,O,\mu_1}, y_i^{R,O,\mu_2}, y_i^{A,H,H,v_1}, y_i^{A,H,H,v_2}, y_i^{A,H,H,v_3}, y_i^{A,H,H,v_4}, y_i^{A,H,O,v_1}, y_i^{A,H,O,v_2}, y_i^{A,H,O,v_3}, y_i^{A,H,O,v_4}, y_i^{A,O,O,v_1}, y_i^{A,O,O,v_2}, y_i^{A,O,O,v_3}, y_i^{A,O,O,v_4}) \quad (2.3.2)$$

3. CODE USE EXAMPLE

The `aevmod` package provides C++ computations of the AEV, its Jacobian, and its Hessian, accessible from Python using `pybind11`. The code provides two classes, both accessible from Python,

- `aev` : contains functionality to define an AEV construction, and evaluate all its associated quantities of interest, including the AEV itself, its Jacobian, and its Hessian.
- `config` : contains functionality to define an atomic configuration, and build its various index sets.

Here's the example code `taev.py` that uses the package. It defines atom types, specifies parameters that define the AEV structure, specifies a CH₂ molecule/configuration, defines two geometries of the CH₂ configuration, then computes the AEVs, Jacobians, and Hessians for each which are written to output text files.

```
1 # A simple test code for computing the AEV and its Jacobian & Hessian
2 import numpy as np
3 import aevmod
4
5 # define types of atom in system
6 types = ['C', 'H']
7
8 # define AEV structure object
9 nrho_rad = 32 # number of radial aev radial shells
10 nrho_ang = 8 # number of angular aev radial shells
11 nalpha = 8 # number of angular aev angular sectors dividing [0,pi]
12 R_c_rad = 4.6 # radial aev cutoff radius (Angstroms)
13 R_c_ang = 3.1 # angular aev cutoff radius (Angstroms)
14 myaev = aevmod.aev(types, nrho_rad, nrho_ang, nalpha, [R_c_rad, R_c_ang])
15 print("built aev object, AEV size:", myaev.dout)
16
17 # define CH2 molecule symbol list
18 symb = ['C', 'H', 'H']
19 print("configuration:", symb)
20
21 # define CH2 molecule object
22 cnf = aevmod.config(symb)
23
24 # build index sets
25 myaev.build_index_sets(cnf)
26 print("radial index sets:", cnf.get_radial_index_set())
27 print("angular index sets:", cnf.get_angular_index_set())
28
29 # define numpy array composed of two CH2 xyz structures
30 vxyz = np.array([[0., 0., 0., 0., 0., 1.10771, 1.08378, 0., -0.22899],
31                 [0., 0., 0., 0., 0., 1.07672, 0.77665, 0., -0.74575]
32                ])
33 print("structures:")
34 with np.printoptions(precision=4, suppress=True):
35     for v in vxyz:
36         print(v)
```



```

37
38 # add vxyz structures to cnf
39 npt = cnf.add_structures(vxyz)
40 print("number of structures:", npt)
41
42 # evaluate AEVs for the array of structures
43 # got_aev[j] is the AEV for structure j
44 # got_aev[j][k] is the AEV for atom k in structure j
45 got_aev = myaev.eval(cnf)
46
47 # write out AEVs
48 fname = "aev.out"
49 print("printing AEVs to", fname)
50 np.savetxt(fname, ["# AEV:"], fmt='%s')
51 with open(fname, "a") as f:
52     for j in range(npt):
53         for k in range(len(symb)):
54             tag="# structure:"+str(j)+", atom:"+symb[k]+"\\n"
55             f.write(tag)
56             np.savetxt(f, got_aev[j][k])
57
58 # evaluate Jacobians of AEVs
59 got_jac = myaev.eval_Jac(cnf)
60
61 # write out Jacobians
62 fname = "jac.out"
63 print("printing Jacobians to", fname)
64 np.savetxt(fname, ["# Jac:"], fmt='%s')
65 with open(fname, "a") as f:
66     for j in range(npt):
67         for k in range(len(symb)):
68             tag="# structure:"+str(j)+", atom:"+symb[k]+"\\n"
69             f.write(tag)
70             np.savetxt(f, got_jac[j][k])
71
72 # evaluate Hessians of AEVs
73 got_hes = myaev.eval_Hess_sac(cnf)
74
75 # write out Hessians
76 fname = "hes.out"
77 print("printing Hessians to", fname)
78 np.savetxt(fname, ["# Hes:"], fmt='%s')
79 with open(fname, "a") as f:
80     for j in range(npt):
81         for k in range(len(symb)):
82             for l in range(myaev.dout):
83                 tag="# structure:"+str(j)+", atom:"+symb[k]+", AEV element:"+str(l)+"\\n"
84                 f.write(tag)
85                 np.savetxt(f, got_hes[j][k][l])

```

Here is some explanation of this example code,

- line 3: Here is where `aevmol` is imported.
- line 6: The AEV structure depends on the atom types in the problem, `types` is a list of atom types. This example includes molecules and/or atomic configurations that contain only C and H atoms.
- lines 9-14: Here we define various parameters that determine the AEV structure, as indicated in the code comments, and then we instantiate an AEV object `myaev` with the structure defined by `types` and the various shown parameters. Note that this object is usable as an AEV for any molecule/configuration comprised of only C and H atoms. The specific case of

one (C,H) molecule or another does not alter the AEV structure, but rather affects the index sets, as explained in subsequent bullets.

- line 18: Here we define a specific configuration comprised of one C atom and two H atoms. The list `symb` simply includes the types of each atom in the configuration, listed in the same order that their Cartesian coordinates are defined below.
- line 22: Here we instantiate a configuration object `cnf` for the configuration `symb`.
- line 25: Here we build index-sets for the configuration `cnf`.
- line 21: The `numpy` array `vxyz` is a 2-dimensional array with each row containing the Cartesian spatial coordinates defining one structure/geometry of the configuration `cnf`. Thus, if the molecule/configuration has N atoms, then `vxyz` needs to have $3N$ entries in each row. Further, `vxyz` can include an arbitrary number of geometries of this configuration, each being a data point. Thus, with `npt` structures (here `npt=2`), `vxyz` is an array with `npt` rows and $3N$ columns.
- line 39: Here we add the structures specified in `vxyz` to the data space of `cnf`, returning `npt` for confirmation.
- line 45: This evaluates the AEV, according to Eq. (2.1.11) above, for each of the atoms in each of the structures defined in `vxyz` and contained in `cnf`. It relies on the `eval` method in the `aev` class.
- line 59: This evaluates the analytical Jacobians, following Eqs. (2.2.1,2.2.3,2.2.5) above, for each of the AEVs for the structures in `cnf`.
- line 73: Here we evaluate the Hessians for each of the AEVs for the structures in `cnf`. We rely on automatic differentiation using Sacado [1]. The Sacado header files are included the `aevmod` distribution under the `include` folder.

REFERENCES

- [1] <https://trilinos.github.io/sacado.html>.
- [2] J. Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.*, 134:074106, 2011.
- [3] J. Behler. Constructing High-Dimensional Neural Network Potentials: A Tutorial Review. *Int. J. Quant. Chem.*, 115:1032–1050, 2015.
- [4] J. Behler. Neural network potential-energy surfaces for atomistic simulations. *Chem. Modell.*, 7:1–41, 2017.
- [5] J. Behler and M. Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98:146401, 2007.
- [6] J. S. Smith, O. Isayev, and A. E. Roitberg. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.*, 8:3192–3203, 2017.

DISTRIBUTION

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
Technical Library	01177	libref@sandia.gov



Sandia
National
Laboratories

Sandia National Laboratories
is a multimission laboratory
managed and operated by
National Technology &
Engineering Solutions of
Sandia LLC, a wholly owned
subsidiary of Honeywell
International Inc., for the U.S.
Department of Energy's
National Nuclear Security
Administration under contract
DE-NA0003525.