SOAPLite Documentation

July 31, 2018

Abstract

SOAP is a state-of-the-art local chemical descriptor that has achieved amazing accuracies in machine learning in chemistry [1,2,3,4]. SoapLite is a fast and lightweight SOAP-spectra [1,2] calculator that gives local chemical descriptors of 1) arbitrary positions in a chemical structure, or 2) every atom position in a chemical structure, both with periodic and non-periodic boundary conditions. SOAPLite was designed for ease-of-use, portability and computational performance in mind, with the potential to perform real-time SOAP analysis. Computation of the kernel is not included, and must be done independently such as with scikit-learn, by implementing from scratch, or by other ready-made SOAP implementations such as with glosim [5].

1 Derivation of SOAP Spectrum

The core algorithm is based on the papers in the reference [1,2]. In SOAPLite, the radial basis $g_{nl}^{(l)}$ and atomic density ρ are

$$g_{nl}^{(l)}(r) = \sum_{k=1}^{N_k} \beta_{nk}^{(l)} r^l e^{-\alpha_{kl} r^2}, \tag{1}$$

$$\rho_{N_A}^{\mu}(\mathbf{r}) = \sum_{i=1}^{N_A} e^{-(\mathbf{r}-\mathbf{r}_i^{\mu})^2} = \sum_{i=1}^{N_A} e^{-(x^2+y^2+z^2-2(xx_i^{\mu}+yy_i^{\mu}+zz_i^{\mu}))} e^{-(x_i^{\mu 2}+y_i^{\mu 2}+z_i^{\mu 2})}. \tag{2}$$

where μ is the type of the atom, N_A is the number of atoms from the center (the desired local position $\mathbf{x}$ of SOAP spectrum translated to the origin), x, y and z are the position in the Cartesian coordinates, α, β give the set of orthonormal radial basis functions and N_k is the number of radial basis functions. The SOAP power spectrum at a certain position $\mathbf{x}$ is defined as

$$\mathbf{P} = P_{nn'l}^{\mu\nu}(\mathbf{x}) = \sum_m c_{nlm}^{\mu}(\mathbf{x}) c_{n'lm}^{*\nu}(\mathbf{x}) \tag{3}$$

where

$$c^{\mu}_{nlm} = \int dV g^{(l)}_{nl}(r)\rho_{N_A}(\mathbf{r})Y_{lm}(\theta,\phi) \tag{4}$$

with different atom types μ, ν and by summing m, the spectra is guaranteed to be rotationally invariant. Y_{lm} are the spherical harmonics.

To find $\beta^{(l)}_{nk}$, we need the analytical formulas

$$\int_0^{\infty} dr\, r^{2(l+1)} e^{-cr^2} = \frac{1}{2}\Gamma\left(l+\frac{3}{2}\right) c^{-l-3/2} \tag{5}$$

$$\int_0^{x} dr\, r^{2(l+1)} e^{-cr^2} = \frac{x^{2l-1}}{2c^2}(cx^2)^{1/2-l}\left(\Gamma\left(l+\frac{3}{2}\right) - \Gamma\left(l+\frac{3}{2}, cx^2\right)\right) \tag{6}$$

where Γ is the Gamma function, for every l, we can build a matrix of $N_k \times N_k$ by integrating over several different α's, we get

$$G^{(l)}_{nn'} = \int_0^{\infty} dr\, r^2\, r^l e^{-\alpha_{nl} r^2} e^{-\alpha_{n'l} r^2} \tag{7}$$

Finally, by inverting $G_{nn'}$ and take the matrix square-root, we systematically get the β

$$\beta^{(l)} = G^{-1/2}_{(l)} \tag{8}$$

In SOAPLite, the α_n's are selected so that it damps at every $(r_{\text{soft\_cut}} + n)/N_k$ with a hard cutoff to avoid unnecessary integrations that will give zero, $r_{\text{hard\_cut}} = r_{\text{soft\_cut}} + 5\text{Å}$.

2 Code Description

2.1 Navigation

2.1.1 Main Directory

`./soapPy.py` is the main program that wraps the source codes. `./genBasis.py` is the basis generator for the soap, which follows the procedure shown in the Derivation of SOAP Spectrum section. `./batchSoapPy.py` is used in case of getting SOAP spectra from a trajectory file. Structs contains structures (`tests/*.xyz` and `tests/*.pdb`) and positions in space (`tests/*.dat`) one can test. tests contains test python executables with periodic and non-periodic cases. `src` contains the source files written in c.

2.1.2 Source Directory

The naming convention in ./src is as follows: `./src/*Py*` requires a main function or a wrapper (since they are precompiled libraries) and `./src/*Cro*` implied that species crossover terms are edabled in $\mathbf{P}^{\mu\nu}$, so without crossover the spectrum will be $\mathbf{P}^{\mu\mu}$. The numbers on the `./src/*Cro*` files indicate how

many species there are (for example, for 6 species, there will be 21 different combinations), which is branched in the python wrapper (`./soapPy.py`) depending on the number of species in the system.

2.2 Source Codes

Although the code seems overwhelming with numbers, they are just preset or precalculated numbers that are used to get $c_{nn'l}$ that are shown in the appendix.

2.2.1 Inputs

The inputs in the `soap()` in the source code are the pointer from the c wrapper in `soapPy.py`

```
double* c
```

Atom positions, and desired positions (`Apos`) for the power spectrum (`Hpos`)

```
double* Apos; double* Hpos;
```

α's, and β's from the basis functions

```
double* alphas; double* betas;
```

numbers for each atom species as an array, r_{cut}, numbers of atoms in the structure, number of species, number of basis functions, L_{max} and number of desired positions for the power spectrum.

```
int* typeNs ;double rCut; int totalAN;
int Nt; int Ns; int lMax; int Hs;
```

2.2.2 Embedded Arrays

Each vectors, matrices and tensors are stored respectively in a memory allocated array with a pointer, and the values are extracted or manipulated by shifting the positions and dimensions. The arrays are: The positions of the atoms in xyz-coordinate

```
double* x = malloc(sizeof(double)*totalAN);
double* y = malloc(sizeof(double)*totalAN);
double* z = malloc(sizeof(double)*totalAN);
```

z^X where z is the z component of the atoms, and r^X where r is the distance from the reference position and $X = 2, 3, 4, 5, 6, 7, 8$

```
double* zX = malloc(sizeof(double)*totalAN);
double* rX = malloc(sizeof(double)*totalAN);
```

Real and Imaginary part of x component and y component in the form $(x+iy)^X$

```
double* ReImX = malloc(2*sizeof(double)*totalAN);
```

precalculated exponents and preset coefficients where 96 was the minimum number coefficients required for up to $L_{\max} = 9$ that isn't 1 in the paranthesis of the right hand side in the appendix's formulas.

```
double* exes = malloc (sizeof(double)*totalAN);
double* preCoef = malloc(96*sizeof(double)*totalAN);
```

precaculated $\alpha/(1+\alpha)$ and $\beta/(1+\alpha)$ (see Appendix)

```
double* bOa = malloc((lMax+1)*NsNs*sizeof(double));
double* aOa = malloc((lMax+1)*Ns*sizeof(double));
```

and finally c_{nlm} and $P_{nn'l}$

```
double* cnnd = malloc(100*Nt*Ns*Hs*sizeof(double));
double* soapMat =
    malloc(Hs*((Ts*(Ts+2))/2)*Ns*Ns*(lMax+1)*sizeof(double));
```

For a non-crossover case, $Ts*(Ts+1)/2$ is replaced with Ts. The detailed coefficients and formulas are in the appendix.

2.2.3 Functions and SOAP Output

Computing $(x+iy)^X$ for $X = 2, 3, 4, 5, 6, 7, 8$. The reason for the many functions is to speed up the multiplications for example $(x+iy)^5 = (x+iy)^3*(x+iy)^2$ where the right-hand side has been already precomputed, is faster than multiplying $(x+iy)$ 5 times. Every odd array position is the imaginary part and the even position (including 0) is the real part. This makes the size of the ReImX arrays $2\times N_A$

```
double* getReIm2(); double* getReIm3();
void getMulReIm(); void getMulDouble()
```

and similarly, for r^X and z^X with $X = 2, 3, 4, 5, 6, 7, 8$

```
double* getRsZs()
```

To filter out the atoms faraway that would just give zeros for the integrations, `Apos` is rearranged by neighborhood and spits out the number of neighbors

```
int getFilteredPos()
```

the functions to precalculate the $\beta/(1+\alpha)$ and $\alpha/(1+\alpha)$ are

```
void getAlphaBeta()
```

and the following function gets c_{nml}

```
int getC()
```

finally, by summing $\sum_m c^{\mu}_{nml} c^{*\nu}_{n'ml}$ with species μ, we have the soapSpectra which is the output $P^{\mu\nu}_{nn'l}$

```
double* soap()
```

The size of **P** is (H_s(N(N+1))/2×((L+1)(T_s(T_s+1)/2), with the crossover and (H_s(N(N+1))/2×((L_{max}+1)Ts) without, where H_s are the number of positions, N is the number of basis functions L is L_{max}, and T_s is the number of species.

3 Tests and Examples

3.1 Symmetry Test

Here, translational and rotational invariance of the SOAP spectra is tested with a $Au_{40}Cu_{40}$ cluster and a SiH_4 molecule by running

```
python test_symmetry.py
```

in the `tests/` directory. For $Au_{40}Cu_{40}$, the structure is rotated 45 degrees around x,y, and z-axis and translated in a few directions and the soap spectra is compared taking the maximum differences. For SiH_4, the soap spectra of hydrogen are compared to each other.

3.2 examples

In the tests directory, there are files called `example_non_periodic_SoapPy.py` and `example_periodic_SoapPy.py`. The procedures are as follows:

(1) Define structure

(2) Define desired positions for SOAP spectra

(3) Set radial basis functions

(4) Get SOAP spectra (CAUTION: Make sure to use the same settings as the set basis function in (3))

(5) Save SOAP spectra in a text file

in 4), we can switch functions between if it is period, non periodic, locals or structures depending on the application. For locals, we need to define positions, while for structures, the positions will be assigned to each atoms in the structure.

The output is a matrix of (Number of defined positions × Descriptor size). Thus, Each row corresponds to one string of soap spectrum. We can compare two soap spectra by taking two rows (within the same matrix or different matrices if the descriptor size is the same).

4 Possible Applications

We can compare two different local chemical environments by taking two soap spectra distances dist($\mathbf{P}_1, \mathbf{P}_2$), whic gives the (dis)similarities. The distance measure can be Euclidean, absolute difference or any distance measure that works for an application. One possible application is finding unique surface atoms (see figures). Aslo, it can be used as an input for a machine learning method to learn physical quantities (by the kernel trick, one can also use it as a structural/global descriptor [2], and machine learn a global quantity such as energy).

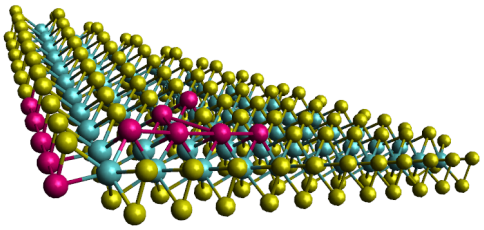

Figure 1: Unique surface atoms on a MoS_2 nanocluster (in pink) by taking the soap differences and eliminating similar local environments with a threashold (dist($\mathbf{P}_1, \mathbf{P}_2$) $<$ threashold). This example gave what a human could guess. In this case, the symmetries of the shape were clear and it wasn't sensitive to the threashold.

5 Possible Improvements and Speedups

Besides eleminating the transpose symmetry of the power spectra, here are possible improvements that can be done on the algorithm to speed of the execution.

1) at least in our system, the majority of the time is spent on computing the exponents. Using a less accurate but a faster exponential function might speed up the code significantly, such as by tabulating the exponents at relevant domains.

2) for every position, the code calculates the distance^2's to filter out faraway atoms that don't contribute to the integration. This is negligible compared to (1) unless the molecar structure is huge. This can be optimized by rearanging the order of the atoms in the input file before, or using other filtering schemes.

3) the radial basis function is produced in python (`genBasis.py`) that gives the α's and β's, however this is only to generalize the process, but not recomended if SOAPLite is embedded in another software for speed, and α's and β's should be tabulated instead.

4) RAM usage is not highly optimized, so if the structure doesn't contain atom numbers in the millions, this shouldn't be a problem, but if a massive

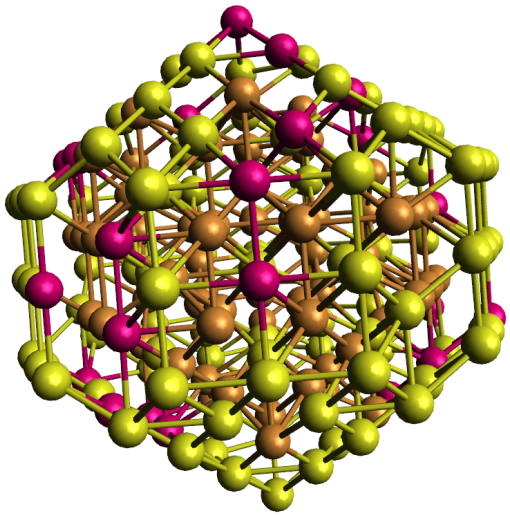

Figure 2: Unique surface atoms on a AuCu nanocluster (in pink) similar to figure. 1. To ask a human to guess unique surface atoms on this system, one will have to take a significant amount of time analysing it (if even possible).

structure is present, RAM usage and (2) is nessesary to be optimized.

If you have any questions, suggestions or comments please contanct:
eiaki.morooka@aalto.fi

References

[1] Albert P. Bartók, Risi Kondor and Gábor Csányi *On representating chemical environments.* https://arxiv.org/abs/1209.3140

[2] Sandi De, Albert Bartók, Gábor Csányi and Michele Ceriotti *Comparing molecules and solids across structural and alchemical space.* https://arxiv.org/abs/1601.04077

[3] Nongnuch Artrith, Alexander Urban and Gerbrand Ceder *Efficient and Accurate Machine-Learning Interpolation of Atomic Energies in Compositions with Many Species.* https://arxiv.org/abs/1706.06293

[4] Albert P. Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R. Kermode, Gábor Csányi, Michele Ceriotti *Machine learning unifies the modeling of materials and molecules* https://arxiv.org/pdf/1706.00179.pdf

[5] https://github.com/cosmo-epfl/glosim

A Integrations

The following formulas are the analytical forms of the integrations for c_{nlm}.

$$c_{n00} = \sum_{k=1}^{3} \frac{\pi\beta_{nk}}{2(1+\alpha_{k0})^{\frac{3}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k0}}{1+\alpha_{k0}} r_i^2}$$

$$c_{n10} = \sum_{k=1}^{N_k} \frac{\sqrt{3}\pi\beta_{nk}}{2(1+\alpha_{k1})^{\frac{5}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k1}}{1+\alpha_{k1}} r_i^2} z_i$$

$$c_{n11} = -\sum_{k=1}^{N_k} \frac{\sqrt{3}\pi\beta_{nk}}{2\sqrt{2}(1+\alpha_{k1})^{\frac{5}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k1}}{1+\alpha_{k1}} r_i^2} (x_i + iy_i)$$

$$c_{n20} = \sum_{k=1}^{N_k} \frac{\sqrt{5}\pi\beta_{nk}}{4(1+\alpha_{k2})^{\frac{7}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k2}}{1+\alpha_{k2}} r_i^2} (3z^2 - r^2)$$

$$c_{n21} = -\sum_{k=1}^{N_k} \frac{\sqrt{15}\pi\beta_{nk}}{2\sqrt{2}(1+\alpha_{k2})^{\frac{7}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k2}}{1+\alpha_{k2}} r_i^2} (x_i + iy_i) z_i$$

$$c_{n22} = \sum_{k=1}^{N_k} \frac{\sqrt{15}\pi\beta_{nk}}{4\sqrt{2}(1+\alpha_{k2})^{\frac{7}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k2}}{1+\alpha_{k2}} r_i^2} (x_i + iy_i)^2$$

$$c_{n30} = \sum_{k=1}^{N_k} \frac{\sqrt{7}\pi\beta_{nk}}{4(1+\alpha_{k3})^{\frac{9}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k3}}{1+\alpha_{k3}} r_i^2} z_i(5z^2 - 3r^2)$$

$$c_{n31} = -\sum_{k=1}^{N_k} \frac{\sqrt{21}\pi\beta_{nk}}{8(1+\alpha_{k3})^{\frac{9}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k3}}{1+\alpha_{k3}} r_i^2} (x_i + iy_i)(4z_i^2 - x_i^2 - y_i^2)$$

$$c_{n32} = \sum_{k=1}^{N_k} \frac{\sqrt{105}\pi\beta_{nk}}{4\sqrt{2}(1+\alpha_{k3})^{\frac{9}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k3}}{1+\alpha_{k3}} r_i^2} (x_i + iy_i)^2 z_i$$

$$c_{n33} = -\sum_{k=1}^{N_k} \frac{\sqrt{35}\pi\beta_{nk}}{8(1+\alpha_{n3})^{\frac{9}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k3}}{1+\alpha_{k3}} r_i^2} (x_i + iy_i)^3$$

$$c_{n40} = \sum_{k=1}^{N_k} \frac{3\pi\beta_{nk}}{16(1+\alpha_{k4})^{\frac{11}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k4}}{1+\alpha_{k4}} r_i^2} (35z_i^4 - 30z_i^2 r_i^2 + 3r_i^4)$$

$$c_{n41} = -\sum_{k=1}^{N_k} \frac{3\sqrt{5}\pi\beta_{nk}}{8(1+\alpha_{k4})^{\frac{11}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k4}}{1+\alpha_{k4}} r_i^2} (x_i + iy_i) z_i (7z_i^2 - 3r_i^2)$$

$$c_{n42} = \sum_{k=1}^{N_k} \frac{3\sqrt{5}\pi\beta_{nk}}{8\sqrt{2}(1+\alpha_{k4})^{\frac{11}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k4}}{1+\alpha_{k4}} r_i^2} (x_i + iy_i)^2 (7z_i^2 - r_i^2)$$

$$c_{n43} = -\sum_{k=1}^{N_k} \frac{3\sqrt{35}\pi\beta_{nk}}{8(1+\alpha_{k4})^{\frac{11}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k4}}{1+\alpha_{k4}} r_i^2} (x_i + iy_i)^3 z_i$$

$$c_{n44} = \sum_{k=1}^{N_k} \frac{3\sqrt{35}\pi\beta_{nk}}{16\sqrt{2}(1+\alpha_{k4})^{\frac{11}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k4}}{1+\alpha_{k4}} r_i^2} (x_i + iy_i)^4$$

$$c_{n50} = \sum_{k=1}^{N_k} \frac{\sqrt{11}\pi\beta_{nk}}{16(1+\alpha_{k5})^{\frac{13}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k5}}{1+\alpha_{k5}} r_i^2} z_i(63z_i^4 - 70z_i^2 r_i^2 + 15r_i^4)$$

$$c_{n51} = -\sum_{k=1}^{N_k} \frac{\sqrt{165}\pi\beta_{nk}}{16\sqrt{2}(1+\alpha_{k5})^{\frac{13}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k5}}{1+\alpha_{k5}} r_i^2} (x_i + iy_i)(21z_i^4 - 14z_i^2 r_i^2 + r_i^4)$$

$$c_{n52} = \sum_{k=1}^{N_k} \frac{\sqrt{1155}\pi\beta_{nk}}{8\sqrt{2}(1+\alpha_{k5})^{\frac{13}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k5}}{1+\alpha_{k5}} r_i^2} (x_i + iy_i)^2 z_i(3z_i^2 - r_i^2)$$

$$c_{n53} = -\sum_{k=1}^{N_k} \frac{\sqrt{385}\pi\beta_{nk}}{32(1+\alpha_{n5})^{\frac{13}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k5}}{1+\alpha_{k5}} r_i^2} (x_i + iy_i)^3(9z_i^2 - r_i^2)$$

$$c_{n54} = \sum_{k=1}^{N_k} \frac{3\sqrt{385}\pi\beta_{nk}}{16\sqrt{2}(1+\alpha_{k5})^{\frac{13}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k5}}{1+\alpha_{k5}} r_i^2} (x_i + iy_i)^4 z_i$$

$$c_{n55} = -\sum_{k=1}^{N_k} \frac{3\sqrt{77}\pi\beta_{nk}}{32(1+\alpha_{k5})^{\frac{13}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k5}}{1+\alpha_{k5}} r_i^2} (x_i + iy_i)^5$$

$$c_{n60} = \sum_{k=1}^{N_k} \frac{\sqrt{13}\pi\beta_{nk}}{32(1+\alpha_{n6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (231z_i^6 - 315z_i^4 r_i^2 + 105z_i^2 r_i^4 - 5r_i^6)$$

$$c_{n61} = -\sum_{k=1}^{N_k} \frac{\sqrt{273}\pi\beta_{nk}}{16\sqrt{2}(1+\alpha_{n6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (x_i + iy_i) z_i(33z_i^4 - 30z_i^2 r_i^2 + 5r_i^4)$$

$$c_{n62} = \sum_{k=1}^{N_k} \frac{\sqrt{1365}\pi\beta_{nk}}{64(1+\alpha_{k6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (x_i + iy_i)^2(33z_i^4 - 18z_i^2 r_i^2 + r_i^4)$$

$$c_{n63} = -\sum_{k=1}^{N_k} \frac{\sqrt{1365}\pi\beta_{nk}}{32(1+\alpha_{k6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (x_i + iy_i)^3 z_i(11z_i^2 - 3r_i^2)$$

$$c_{n64} = \sum_{k=1}^{N_k} \frac{3\sqrt{91}\pi\beta_{nk}}{32\sqrt{2}(1+\alpha_{k6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (x_i + iy_i)^4(11z_i^2 - r_i^2)$$

$$c_{n6-5} = -\sum_{k=1}^{N_k} \frac{3\sqrt{1001}\pi\beta_{nk}}{32(1+\alpha_{n6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (x_i + iy_i)^5 z_i$$

$$c_{n6-6} = \sum_{k=1}^{N_k} \frac{\sqrt{3003}\pi\beta_{nk}}{64(1+\alpha_{k6})^{\frac{15}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k6}}{1+\alpha_{k6}} r_i^2} (x_i + iy_i)^6$$

$$c_{n70} = \sum_{k=1}^{N_k} \frac{\sqrt{15}\pi\beta_{nk}}{32(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} z_i(429z_i^6 - 693z_i^4 r_i^2 + 315z_i^2 r_i^4 - 35r_i^6)$$

$$c_{n71} = -\sum_{k=1}^{N_k} \frac{\sqrt{105}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)(429z_i^6 - 495z_i^4 r_i^2 + 135z_i^2 r_i^4 - 5r_i^6)$$

$$c_{n72} = \sum_{k=1}^{N_k} \frac{3\sqrt{35}\pi\beta_{nk}}{64(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)^2 z_i(143z_i^4 - 110z_i^2 r_i^2 + 15r_i^4)$$

$$c_{n73} = -\sum_{k=1}^{N_k} \frac{3\sqrt{35}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)^3(143z_i^4 - 66z_i^2 r_i^2 + 3r_i^4)$$

$$c_{n74} = \sum_{k=1}^{N_k} \beta_{nk} \frac{3\sqrt{385}\pi\beta_{nk}}{32\sqrt{2}(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)^4 z_i(13z_i^2 - 3r_i^2)$$

$$c_{n75} = -\sum_{k=1}^{N_k} \frac{3\sqrt{385}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)^5(13z_i^2 - r_i^2)$$

$$c_{n76} = \sum_{k=1}^{N_k} \frac{3\sqrt{5005}\pi\beta_{nk}}{64(1+\alpha_{n7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)^6 z_i$$

$$c_{n77} = -\sum_{k=1}^{N_k} \frac{3\sqrt{715}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k7})^{\frac{17}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k7}}{1+\alpha_{k7}} r_i^2} (x_i + iy_i)^7$$

$$c_{n80} = \sum_{k=1}^{N_k} \frac{\sqrt{17}\pi\beta_{nk}}{256(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (6435 z_i^8 - 12012 z_i^6 r_i^2 + 6930 z_i^4 r_i^4 - 1260 z_i^2 r_i^6 + 35 r_i^8)$$

$$c_{n81} = -\sum_{k=1}^{N_k} \frac{3\sqrt{17}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i) z_i (715 z_i^6 - 1001 z_i^4 r_i^2 + 385 z_i^2 r_i^4 - 35 r_i^6)$$

$$c_{n82} = \sum_{k=1}^{N_k} \frac{3\sqrt{595}\pi\beta_{nk}}{128(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^2 (143 z_i^6 - 143 z_i^4 r_i^2 + 33 z_i^2 r_i^4 - r_i^6)$$

$$c_{n83} = -\sum_{k=1}^{N_k} \frac{\sqrt{19635}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^3 z_i (39 z_i^4 - 26 z_i^2 r_i^2 + 3 r_i^4)$$

$$c_{n84} = \sum_{k=1}^{N_k} \frac{3\sqrt{1309}\pi\beta_{nk}}{128\sqrt{2}(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^4 (65 z_i^4 - 26 z_i^2 r_i^2 + r_i^4)$$

$$c_{n85} = -\sum_{k=1}^{N_k} \frac{3\sqrt{17017}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^5 z_i (5 z_i^2 - r_i^2)$$

$$c_{n86} = \sum_{k=1}^{N_k} \frac{\sqrt{7293}\pi\beta_{nk}}{128(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^6 (15 z_i^2 - r_i^2)$$

$$c_{n8-7} = -\sum_{k=1}^{N_k} \frac{3\sqrt{12155}\pi\beta_{nk}}{64\sqrt{2}(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^7 z_i$$

$$c_{n88} = \sum_{k=1}^{N_k} \frac{3\sqrt{12155}\pi\beta_{nk}}{256\sqrt{2}(1+\alpha_{k8})^{\frac{19}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k8}}{1+\alpha_{k8}} r_i^2} (x_i + iy_i)^8$$

$$c_{n90} = \sum_{k=1}^{N_k} \frac{\sqrt{19}\pi\beta_{nk}}{256(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} z_i (12155 z_i^8 - 25740 z_i^6 r_i^2 + 18018 z_i^4 r_i^4 - 4620 z_i^2 r_i^6 + 315 r_i^8)$$

$$c_{n91} = -\sum_{k=1}^{N_k} \frac{3\sqrt{95}\pi\beta_{nk}}{256\sqrt{2}(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i) (2431 z_i^8 - 4004 z_i^6 r_i^2 + 2002 z_i^4 r_i^4 - 308 z_i^2 r_i^6 + 7 r_i^8)$$

$$c_{n92} = \sum_{k=1}^{N_k} \frac{3\sqrt{1045}\pi\beta_{nk}}{128(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^2 z_i (221 z_i^6 - 273 z_i^4 r_i^2 + 91 z_i^2 r_i^4 - 7 r_i^6)$$

$$c_{n93} = -\sum_{k=1}^{N_k} \frac{\sqrt{21945}\pi\beta_{nk}}{256(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^3 (221 z_i^6 - 195 z_i^4 r_i^2 + 39 z_i^2 r_i^4 - r_i^6)$$

$$c_{n94} = \sum_{k=1}^{N_k} \frac{3\sqrt{95095}\pi\beta_{nk}}{128\sqrt{2}(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^4 z_i (17 z_i^4 - 10 z_i^2 r_i^2 + r_i^4)$$

$$c_{n95} = -\sum_{k=1}^{N_k} \frac{3\sqrt{2717}\pi\beta_{nk}}{256(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^5 (85 z_i^4 - 30 z_i^2 r_i^2 + r_i^4)$$

$$c_{n96} = \sum_{k=1}^{N_k} \frac{\sqrt{40755}\pi\beta_{nk}}{128(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^6 z_i (17 z_i^2 - 3 r_i^2)$$

$$c_{n97} = -\sum_{k=1}^{N_k} \frac{3\sqrt{13585}\pi\beta_{nk}}{512(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^7 (17 z_i^2 - r_i^2)$$

$$c_{n98} = \sum_{k=1}^{N_k} \frac{3\sqrt{230945}\pi\beta_{nk}}{256\sqrt{2}(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^8 z_i$$

$$c_{n99} = -\sum_{k=1}^{N_k} \frac{\sqrt{230945}\pi\beta_{nk}}{512(1+\alpha_{k9})^{\frac{21}{2}}} \sum_{i=1}^{N_A} e^{-\frac{\alpha_{k9}}{1+\alpha_{k9}} r_i^2} (x_i + iy_i)^9$$