

# Using van der Waals Density Functionals in ONETEP

Lampros Andrinopoulos  
Imperial College London

November 2012

## Activating vdW-DF

The van der Waals energy is calculated in ONETEP using the van der Waals Density Functional method, developed by Dion *et al* [1].

The only input variable needed to activate the vdW-DF functional is to set `xc_functional` `VDWDF`. If a `vdw_df_kernel` file is not present in the working directory, it will be automatically generated.

## Theory

The form for the exchange-correlation functional proposed by Dion *et al* is:

$$E_{xc} = E_x^{\text{revPBE}} + E_c^{\text{PW92}} + E_c^{\text{nl}} \quad (1)$$

where the non-local exchange-correlation energy is given by:

$$E_c^{\text{nl}} = \frac{1}{2} \int \int d\mathbf{r} d\mathbf{r}' \rho(\mathbf{r}) \phi(\mathbf{r}, \mathbf{r}') \rho(\mathbf{r}') \quad (2)$$

where  $\rho(\mathbf{r})$  is the electron density at  $\mathbf{r}$  and  $\phi(\mathbf{r}, \mathbf{r}')$  is the nonlocal exchange correlation kernel whose form is explained in [1].

## Non-local correlation energy

The direct calculation of the integral in the form of Eq. 2 is very computationally expensive, as it involves a six-dimensional spatial integral.

The algorithm proposed later by Roman-Perez and Soler [2] improves the efficiency of the calculation. They observed that with the form used by Dion *et al* for  $\phi$ , the above expression can be re-written as:

$$E_c^{\text{nl}} = \frac{1}{2} \int \int d\mathbf{r} d\mathbf{r}' \rho(\mathbf{r}) \phi(q, q', r) \rho(\mathbf{r}') \quad (3)$$

where  $r = |\mathbf{r} - \mathbf{r}'|$ , and  $q$  and  $q'$  are the values of a universal function  $q_0[\rho(\mathbf{r}), |\nabla\rho(\mathbf{r})|]$  at  $\mathbf{r}$  and  $\mathbf{r}'$ . They thus proposed a way to expand the kernel  $\phi$  using interpolating polynomials

$p_\alpha(q)$  for chosen values  $q_\alpha$  of  $q$ , and tabulated functions  $\phi_{\alpha\beta}(r)$  for the kernel corresponding to each pair of interpolating polynomials. The interpolating polynomials  $p_\alpha$  are cubic splines that evaluate to a Kronecker delta on each respective interpolating point. A mesh of 20 interpolation points is used in Soler's implementation. The Soler form of the nonlocal energy can be written as:

$$\phi(q_1, q_2, r) = \sum_{\alpha\beta} \phi_{\alpha\beta}(r) p_\alpha(q_1) p_\beta(q_2) \quad (4)$$

The universal function  $q_0(\mathbf{r})$  is in practice given by:

$$q_0(\mathbf{r}) = \left( 1 + \frac{\epsilon_c^{\text{PW92}}}{\epsilon_x^{\text{LDA}}} + \frac{0.8491}{9} \left( \frac{|\nabla\rho|}{2\rho k_F} \right)^2 \right) k_F \quad (5)$$

with  $k_F = (3\pi^2\rho)^{1/3}$ . The quantity  $q_0$  is first "saturated" to limit its maximum value, according to:

$$q_0^{\text{sat}}(\rho, |\nabla\rho|) = q_c \left( 1 - \exp \left( - \sum_{m=1}^{m_c} \frac{(q/q_c)^m}{m} \right) \right) \quad (6)$$

where  $q_c$  is the maximum value of the mesh of  $q_\alpha$ .

To evaluate this, we first define a quantity  $\theta_\alpha(\mathbf{r}) = \rho(\mathbf{r}) p_\alpha(q(\rho(\mathbf{r}), |\nabla\rho(\mathbf{r})|))$  in real space. In terms of this, Eq.2 can be written as:

$$E_c^{\text{nl}} = \frac{1}{2} \sum_{\alpha\beta} \int \int d\mathbf{r} d\mathbf{r}' \theta_\alpha(\mathbf{r}) \theta_\beta(\mathbf{r}') \phi_{\alpha\beta}(r) \quad (7)$$

It can be shown that this can be written as a reciprocal space integral:

$$E_c^{\text{nl}} = \frac{1}{2} \sum_{\alpha\beta} \int d\mathbf{k} \theta_\alpha^*(\mathbf{k}) \theta_\beta(\mathbf{k}) \phi_{\alpha\beta}(k) \quad (8)$$

Since the kernel is radially dependent in real space, it is only dependent on the magnitude of the G-vectors, hence the kernel need only be evaluated as a 1-dimensional function  $\phi_{\alpha\beta}(k)$  for each  $\alpha, \beta$ .

The kernel  $\phi$  and its second derivatives are tabulated for a specific set of radial points and transformed to reciprocal space. These values are then used to interpolate the kernel at every point  $\mathbf{k}$  in reciprocal space required to calculate Eq. 8.

## Kernel

This section details the evaluation of the NLXC kernel. The kernel  $\phi(\mathbf{r}, \mathbf{r}')$  as specified by Dion *et al* [1] is given by (in atomic units):

$$\phi(\mathbf{r}, \mathbf{r}') = \frac{1}{\pi^2} \int_0^\infty a^2 da \int_0^\infty b^2 db W(a, b) T(\nu(a), \nu(b), \nu'(a), \nu'(b)) \quad (9)$$

where

$$T(w, x, y, z) = \frac{1}{2} \left[ \frac{1}{w+x} + \frac{1}{y+z} \right] \left[ \frac{1}{(w+y)(x+z)} + \frac{1}{(w+z)(y+x)} \right], \quad (10)$$

and

$$W(a, b) = 2 \left[ (3 - a^2)b \cos b \sin a \right. \quad (11)$$

$$\left. + (3 - b^2)a \cos a \sin b \right. \quad (12)$$

$$\left. + (a^2 + b^2 - 3) \sin a \sin b \right. \quad (13)$$

$$\left. - 3ab \cos a \cos b \right] / (a^3 b^3), \quad (14)$$

and

$$v(y) = 1 - e^{-\gamma y^2/d^2}; \quad v'(y) = 1 - e^{-\gamma y^2/d'^2}; \quad (15)$$

where  $d = |\mathbf{r} - \mathbf{r}'|q_0(\mathbf{r})$ ,  $d' = |\mathbf{r} - \mathbf{r}'|q_0(\mathbf{r}')$

Following this chain of logic, it is clear that this the kernel can in fact be considered as a function only of  $|\mathbf{r} - \mathbf{r}'|$ ,  $q_0(\mathbf{r})$  and  $q_0(\mathbf{r}')$ , since all other variables are dummy variables which are integrated over. The kernel can therefore be written as

$$\phi(r, q_0(\mathbf{r}), q_0(\mathbf{r}')) \quad (16)$$

This makes it possible to evaluate the integrals above so as to tabulate the kernel values numerically for a pre-chosen set of radial points and  $q_0$  values.

## Non-local potential

Starting from Eq. 8, one can evaluate the potential  $v^{\text{nl}}(\mathbf{r})$  corresponding to this energy, by evaluating all terms in  $\partial E_{\text{nl}}/\partial n(\mathbf{r})$ . The non-local potential  $v_i^{\text{nl}}$  at point  $\mathbf{r}_i$  on the grid is thus written explicitly in terms of the derivatives of the  $\theta_\alpha$  quantities with respect to the values  $\rho_j$  at all other points on the grid:

$$v_i^{\text{nl}} = \sum_{\alpha} \left( u_{\alpha i} \frac{\partial \theta_{\alpha i}}{\partial \rho_i} + \sum_j u_{\alpha j} \frac{\partial \theta_{\alpha j}}{\partial \nabla \rho_j} \frac{\partial \nabla \rho_j}{\partial \rho_i} \right) \quad (17)$$

This makes use of the quantities  $u_{\alpha}(\mathbf{r}) = \sum_{\beta} \mathcal{F}(\theta_{\beta}(\mathbf{k})\phi_{\alpha\beta}(k))$ : which are already calculated in the evaluation of the energy.

Using the White and Bird [3] approach, Eq. 17 can be written as:

$$v_{\text{nl}}(\mathbf{r}) = \sum_{\alpha} \left( u_{\alpha}(\mathbf{r}) \frac{\partial \theta_{\alpha}(\mathbf{r})}{\partial \rho(\mathbf{r})} - \int \int i\mathbf{G} \cdot \frac{\nabla \rho(\mathbf{r}')}{|\nabla \rho(\mathbf{r}')|} \frac{\partial \theta_{\alpha}(\mathbf{r}')}{\partial |\nabla \rho(\mathbf{r}')|} e^{i\mathbf{G} \cdot (\mathbf{r} - \mathbf{r}')} d\mathbf{r} d\mathbf{G} \right) \quad (18)$$

For this we need to calculate  $\frac{\partial \theta}{\partial \rho}$  and  $\frac{\partial \theta}{\partial |\nabla \rho|}$ :

$$\begin{aligned}
\frac{\partial \theta_\alpha}{\partial \rho} &= p_\alpha + \rho \frac{\partial p_\alpha}{\partial \rho} \\
&= p_\alpha + \rho \frac{\partial p_\alpha}{\partial q} \frac{\partial q}{\partial \rho} \\
&= p_\alpha + \rho \frac{\partial p_\alpha}{\partial q} \frac{q}{k_F} \frac{\partial k_F}{\partial \rho} + \rho \frac{\partial p_\alpha}{\partial q} k_F \left( \frac{\partial \varepsilon_c}{\partial \rho} \varepsilon_x^{-1} - \varepsilon_c \varepsilon_x^{-2} \frac{\partial \varepsilon_x}{\partial \rho} - \frac{8}{3(3\pi^2)^{2/3}} \frac{Z}{4} (\nabla \rho)^2 \rho^{-11/3} \right) \\
&= p_\alpha + q/3 \frac{\partial p_\alpha}{\partial q} + k_F \rho \frac{\partial p_\alpha}{\partial q} \left( \frac{\partial \varepsilon_c}{\partial \rho} \varepsilon_x^{-1} - \varepsilon_c \varepsilon_x^{-2} \frac{\partial \varepsilon_x}{\partial \rho} - \frac{2Z}{3(3\pi^2)^{2/3}} |\nabla \rho|^2 \rho^{-11/3} \right) \quad (19)
\end{aligned}$$

$$\frac{\partial \theta_\alpha}{\partial |\nabla \rho|} = \rho \frac{\partial p_\alpha}{\partial q} \frac{\partial q}{\partial |\nabla \rho|} = \frac{Z}{2\rho k_F} \rho \frac{\partial p_\alpha}{\partial q} |\nabla \rho| \quad (20)$$

Combining Eqs. 18, 19 and 20 gives us the final expression for the nonlocal potential.

## Overview of computational algorithm

### Module `nlxc_mod`

The main module for the calculation of the non-local energy and potential is `nlxc_mod`. The tabulation of the kernel  $\phi$  is performed only if a kernel file is not found, by `vdwdf_kernel`.

The input required to calculate the non-local energy and potential is essentially just the density and its gradient on the fine grid. The calculation of  $q$  and the Fourier transformed  $\theta_\alpha$  from Eq. 8 is performed first, in the routine `nlxc_q0_theta`. The derivatives of the  $\theta_\alpha$ s with respect to the density and the module of its gradient are calculated on-the-fly in the real-space loop for the calculation of the non-local potential  $v_n l$  in Eq. 17. This is to avoid storing unnecessary arrays. From Eq. 18 two transforms are required per  $\alpha$  value, a forward FFT, followed by a backward FFT for calculating the non-local potential.

Subroutines to interpolate the polynomials as well as the kernel using cubic splines are used (`spline_interp` and `interpolate`). The interpolating polynomials  $p_\alpha$  used are Kronecker deltas, so they take the value 1 on the interpolating point and are zero at the other points.

### Module `vdwdf_kernel`

The kernel  $\phi_{\alpha\beta}(k)$  is tabulated for 1024 radial reciprocal space points and 20  $q_0$  points. Gaussian quadrature is used to calculate Eq. 16 and then the result is Fourier transformed. The second derivatives of the kernel are calculated by interpolation, and also tabulated. The default name of the file is `vdw_df_kernel`. The program will first check if this file exists: if it does, it will be loaded in and need not be calculated. If it does not, it will be generated from scratch (which only takes a few minutes) and then it is written out for future re-use in the current working directory.

The format of the `vdw_df_kernel` file is:  
`N_alpha N_radial`  
`max_radius`  
`q_points(:)`  
`kernel(0:N_radial,alpha,beta)`  
`kernel2(0:N_radial,alpha,beta)`  
where `kernel2` is the array of second derivatives of the kernel):

## References

- [1] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, *Phys. Rev. Lett.* **92**, 246401 (2004).
- [2] G. Román-Pérez and J. M. Soler, *Phys. Rev. Lett.* **103**, 096102 (2009).
- [3] J. A. White and D. M. Bird, *Phys. Rev. B* **50**, 4954 (1994).