Explanation of Jastrow factor (not for distribution)

C. Filippi

The subroutine I am sending contains a simplified version of the Jastrow factor Cyrus and I normally use. The form I am sending has the same variational freedom as the old Jastrow factor published in our JCP paper of 1996.

The new form was motivated by the finding that the *ee* and the *en* terms optimized for an atom are transferable to larger molecules (in particular, I was playing with Si clusters of increasing size) while the *een* terms are not. Therefore, we decided to separate the purely *en* and *ee* terms from the *een* terms (they were mixed in the 1996 Jastrow factor) and Cyrus tried to render the *een* terms transferable by rewriting them in terms of variables which can also be short range. Unfortunately, so far, we have not been able to realize trasferability of the *een* terms.

Cyrus and I are planning to publish the new form of Jastrow factor with all the modifications for periodic systems but, as usual, we have not done it yet.

The subroutine I am sending has been simplified and cleaned from lots of features (Fock's terms, periodic stuff, multiple types of scaling, multiple wave functions etc.). Hopefully, it still works!

The scaling function is given by:

scaling function R for J_{en}, J_{ee}	scaling function R for J_{een}
$\frac{1-e^{-\kappa r}}{\kappa}$	$e^{-\kappa r}$

The Jastrow factor J is written as

$$J = J_{en} J_{ee} J_{een} = \exp(f_{en} + f_{ee} + f_{een}) \tag{1}$$

where

$$f_{en}(R_{i\alpha}) = \sum_{i=1}^{N_{\rm elec}} \sum_{\alpha=1}^{N_{\rm nuc}} \left(\frac{a_1 R_{i\alpha}}{1 + a_2 R_{i\alpha}} + \sum_{p=2}^{N^a_{\rm ord}} a_{p+1} R_{i\alpha}^p \right) \tag{2}$$

$$f_{ee}(R_{ij}) = \sum_{i=2}^{N_{\rm elec}} \sum_{j=1}^{i-1} \left(\frac{b_1 R_{ij}}{1 + b_2 R_{ij}} + \sum_{p=2}^{N^b_{\rm ord}} b_{p+1} R_{ij}^p \right) \tag{3}$$

$$f_{een}(R_{i\alpha}, R_{j\alpha}, R_{ij}) = \sum_{i=2}^{N_{\rm elec}} \sum_{j=1}^{i-1} \sum_{\alpha=1}^{N_{\rm nuc}} \sum_{p=2}^{N^c_{\rm ord}} \sum_{k=p-1}^{0} \sum_{l=l_{\rm max}}^{0} c_n R_{ij}^k (R_{i\alpha}^l + R_{j\alpha}^l)(R_{i\alpha} R_{j\alpha})^m, \quad \text{where } m = \tfrac{p-k-l}{2} \tag{4}$$

and $l_{\rm max}$ is $p-k$ if $k \neq 0$ and $p-k-2$ if $k=0$. Only terms for which $m = \frac{p-k-l}{2}$ is an integer are included. The n in c_n is just used to have a single index, rather than a triple index p,k,l or m,k,l for the c coefficients. Obviously the last two sums could be rewritten so that the increment is 1 rather than -1, but it is written in this form to have the order of the coefficients, c_n, be the same as those in the program.

The a and c coefficients depend on the atom type (extra index *it* in the subroutine). The b coefficients can depend on whether the electron pair has antiparallel or parallel spins but I removed this feature from the subroutine I am sending. The only spin dependence is in b_1 which is is used to satisfy the *ee* cusp conditions (1/2, 1/4 for antiparallel, parallel electrons). b_1 is modified directly inside the subroutine.

The c terms do not contribute to the cusp conditions.

Other information:

1. You need to call `set_scale_dist` once at the beginning of the run.
2. You need to compute the arrays in the common block */distance/* before calling the subroutine:
   $rvec\_en(3, MELEC, MCENT)$ distance vector of each electron from each nucleus
   $r\_en(MELEC, MCENT)$ norm of $rvec\_en(3, MELEC, MCENT)$
   $rvec\_ee(3, MMAT\_DIM2)$ distance vector for electron-electron (for the index, see its use in jastrow4)
   $r\_ee(MMAT\_DIM2)$ norm of $rvec\_ee(3, MMAT\_DIM2)$.

3. The call to jastrow4 sets up the whole Jastrow factor. The quantities in the common block /*jaso*/ are the ones you need to upgrade if you do a single-electron move.

4. In output from jastrow4 are:
   $v(3, MELEC) = \nabla \ln(J)$
   $d2 = \nabla^2 \ln(J)$
   $value = \ln(J)$