Supplemental material for "On-the-fly determination of active region centers in adaptive-partitioning QM/MM"

Zeng-hui Yang*,†,‡

Microsystem and Terahertz Research Center, China Academy of Engineering Physics, Chengdu, China 610200, and Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang, China 621000

E-mail: yangzenghui@mtrc.ac.cn

Abstract

This supplemental material contains the implementation details for the AC-AP method, and the functional form and the derivation of the smoothing function f in Eq. (4) of the main text.

1 Implementation details of the AC-AP method

The AC-AP method is currently implemented in the LAMMPS[1,2] code in C++. It contains two parts: the class for the method and the classes for calculating the criterion properties. Both are derived from `class Fix` by public inheritance. The current implementation only supports the Verlet integrator (`run_style verlet`). It should be noted that the AC-AP method in itself only produces the atomic weights and their derivatives, and a compatible energy-based AP-QM/MM method is required to utilize this information in actual simulations. In the main text, I used the modSISPA[3,4] AP-QM/MM method, and its description and pseudocodes can be found in the reference.[3]

The main part of the AC-AP method is implemented by defining the virtual function `pre_force()` inherited from `class Fix`, which is called by the time integrator after the exchange of atoms between processors (refer to `verlet.cpp`). The purpose of `pre_force()` is to calculate the weights (λ) of all atoms, and to calculate $\partial\lambda_\beta/\partial R_\alpha$ which is the quantity in the parenthesis of Eq. (7) in the main text. The pseudocode for `pre_force()` is shown in the following.

procedure PRE_FORCE($\{\xi_\alpha\}, \left\{\frac{\partial\xi_\alpha}{\partial\vec{R}_\beta}\right\}$)
  $W^{\text{buf}} \leftarrow -\text{RQM}(\xi_{\min}^{\text{semi}})$
  for each α
    do $R_\alpha^{\text{QM}} \leftarrow \text{RQM}(\xi_\alpha)$
  $C \leftarrow$ GENERATECENTERS($\{\xi_\alpha\}$)
  QM, Buf $\leftarrow$ SETUPGROUPS(C)
  $\{\lambda_\alpha\} \leftarrow$ CALCULATEWEIGHTS(QM, Buf, C)
  $\left\{\frac{\partial\lambda_\alpha}{\partial\vec{R}_\beta}\right\} \leftarrow$ CALCULATEDERIV(Buf, C, $\left\{\frac{\partial\xi_\alpha}{\partial\vec{R}_\beta}\right\}$)
  return $(\{\lambda_\alpha\}, \left\{\frac{\partial\lambda_\alpha}{\partial R_\beta}\right\})$

procedure GENERATECENTERS($\{\xi_\alpha\}$)
  $C \leftarrow \{\}$
  for each α
    do if $\xi_\alpha > \xi_{\min}^{\text{semi}}$
    then $C \leftarrow C \cup \{\alpha\} \cup$ PERIODICIMAGE(α)
  return (C)

procedure PERIODICIMAGE(ζ)
  return (periodic images of atom ζ)

*To whom correspondence should be addressed
†Microsystem and Terahertz Research Center, China Academy of Engineering Physics, Chengdu, China 610200
‡Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang, China 621000

procedure SETUPQMGROUP(C)
 QM $\leftarrow \{\}$
 Buf $\leftarrow \{\}$
 for each α
  do for each $\zeta \in C$
  do { **if** $\left|\vec{R}_\alpha - \vec{R}_\zeta\right| < R_\zeta^{\mathrm{QM}}$
  then QM $\leftarrow$ QM $\cup \{\alpha\}$
  else if $\left|\vec{R}_\alpha - \vec{R}_\zeta\right| < R_\zeta^{\mathrm{QM}} + W^{\mathrm{buf}}$
  then Buf $\leftarrow$ Buf $\cup \{\alpha\}$
 return (QM, Buf)

procedure CALCULATEWEIGHTS(QM, Buf, C)
 for each α
  do $\lambda_\alpha \leftarrow 0$
 for each $\alpha \in$ QM
  do $\lambda_\alpha \leftarrow 1$
 for each $\alpha \in$ Buf
  do { $t \leftarrow 1$
  for each $\zeta \in C$
  do if $\left|\vec{R}_\alpha - \vec{R}_\zeta\right| < R_\zeta^{\mathrm{QM}} + W^{\mathrm{buf}}$
  then $t \leftarrow (1 - \lambda_{\alpha,\zeta})\, t$
  $\lambda_\alpha \leftarrow 1 - t$
 return $(\{\lambda_\alpha\})$

procedure CALCULATEDERIV(Buf, C, $\left\{\frac{\partial \xi_\alpha}{\partial \vec{R}_\beta}\right\}$)
 for each α
  do for each β
  do { $\frac{\partial \lambda_\alpha}{\partial \vec{R}_\beta} \leftarrow 0$
  $\frac{\partial \lambda_\alpha}{\partial \xi_\beta} \leftarrow 0$
 for each $\alpha \in$ Buf
  do for each $\zeta \in C$
  do if $\left|\vec{R}_\alpha - \vec{R}_\zeta\right| \in (R_\zeta^{\mathrm{QM}}, R_\zeta^{\mathrm{QM}} + W^{\mathrm{buf}}]$
  then { **if** $\alpha \neq \zeta$
  then { $\vec{t} \leftarrow \frac{\partial \lambda_{\alpha,\zeta}}{\partial \tilde{\lambda}_{\alpha,\zeta}} \frac{\vec{R}_\alpha - \vec{R}_\zeta}{\left|\vec{R}_\alpha - \vec{R}_\zeta\right| W^{\mathrm{buf}}}$
  $\frac{\partial \lambda_\alpha}{\partial \vec{R}_\zeta} \leftarrow \vec{t}$
  $\frac{\partial \lambda_\alpha}{\partial \vec{R}_\alpha} \leftarrow \frac{\partial \lambda_\alpha}{\partial \vec{R}_\alpha} - \vec{t}$
  $\frac{\partial \lambda_\alpha}{\partial \xi_\zeta} \leftarrow \frac{\partial \lambda_{\alpha,\zeta}}{\partial \tilde{\lambda}_{\alpha,\zeta}} \frac{\partial \tilde{\lambda}_{\alpha,\zeta}}{\partial R_\zeta^{\mathrm{QM}}} \frac{\partial R_\zeta^{\mathrm{QM}}}{\partial \xi_\zeta}$
  for each β
  do $\frac{\partial \lambda_\alpha}{\partial R_\beta} \leftarrow \frac{\partial \lambda_\alpha}{\partial R_\beta} + \frac{\partial \lambda_\alpha}{\partial \xi_\zeta} \frac{\partial \xi_\zeta}{\partial R_\beta}$
 return $\left(\left\{\frac{\partial \lambda_\alpha}{\partial \vec{R}_\beta}\right\}\right)$

The structure of the pseudocode is straightforward, and most of the contents are explained in the main text. The part that does not occur in the main text is the treatment of periodicity (shown in Fig. 1), where the periodic images of the QM centers are considered as QM centers as well. It should be noted that for the QM calculation, one has to rearrange the geometry by moving the QM and buffer atoms that belongs to the periodic images of centers. An example of such rearrangement is shown in Fig. 1(b) and 1(c), where the structure in Fig. 1(c) is obtained from rearranging the atoms in Fig. 1(b). The rearrangement is unnecessary for the main text since I used MM/MM for demonstration, where the periodicity in MM calculations are treated by the LAMMPS code.

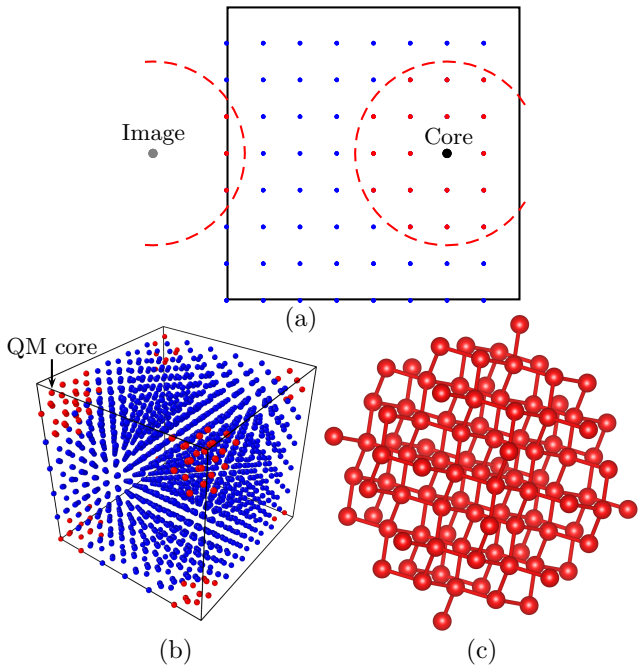

Figure 1: Illustration of the treatment of periodicity. Panel (a) shows that the periodic images of a QM center (denoted as 'core' on figure) are also treated as a center. Panel (b) is an example of bulk Si with an atom at the corner of the supercell chosen as the center. Panel (c) shows the rearranged geometry that is used in the QM calculation.

The actual storage of data involved in `pre_force()` is described below. I use an integer flag to designate whether an atom belongs to the QM, buffer or MM group, and this flag is

stored using the `fix property/atom` provided in LAMMPS. The weights $\{\lambda_\alpha\}$ is also stored this way. I use `std::map` for the derivatives, and the index is `std::pair<tagint,tagint>` since two atoms are involved. There is no guarantee that the two atoms are on the same process, so the maps of the derivatives contain all the data instead of only those local to the process. Only the non-zero entries of the derivatives are stored.

It should be noted that the data in `fix property/atom` only contains the atoms and ghost atoms local to the current process (where ghost atoms represent atoms on other processes or periodic images), so the scaling factor calculation only runs through the local atoms. Since `pre_force()` is called after the exchange of atoms between processes, one has to manually make sure that the ghost atoms have the updated scaling factors by calling `comm->exchange()` and `comm->borders()`. Since the calculation of the criterion property may need updated neighbor list and has to happen before the calculation of $\{\lambda_\alpha\}$, AC-AP cannot be implemented in `pre_exchange()` instead, leading to this complication.

The calculation of the criterion property is implemented similarly, where the class is derived from `class Fix` and the calculation is implemented in `pre_force()`. The criterion property is stored inside the fix to allow access from the LAMMPS input script as an one dimensional array. $\partial\xi_\alpha/\partial\vec{R}_\beta$ is stored using `std::map` similarly as the derivatives of the weights.

2 Smoothing function of the QM region radius

$f(\xi)$ in Eq. (4) of the main text is a monotonically increasing continuous function that satisfies $f(\xi^{\rm QM}_{\rm max}) = 1$ and $f(\xi^{\rm QM}_{\rm min}) = 0$. In practice, I only need to evaluate $f(\xi)$ for $\xi \in [\xi^{\rm semi}_{\rm min}, \xi^{\rm QM}_{\rm max}]$. I obtain a smoothly changing $f(\xi)$ as the following. Define $\tau'(\tau)$ as

$$\tau'(\tau) = \frac{\tau - \tau^{\rm semi}_{\rm min}}{1 - \tau^{\rm semi}_{\rm min}}, \tag{1}$$

where $\tau(\xi) = (\xi - \xi^{\rm QM}_{\rm min})/(\xi^{\rm QM}_{\rm max} - \xi^{\rm QM}_{\rm min})$, and $\tau^{\rm semi}_{\rm min} = \tau(\xi^{\rm semi}_{\rm min})$. I write $f(\xi)$ as $f(\xi) = g(\tau'(\tau(\xi)))$, and g needs to satisfy the following conditions so that the first and second order derivatives of Eq. (4) of the main text are continuous:

$$\begin{array}{ll} g(\tau'(0)) = 0, & g(1) = 1, \\ \left.\frac{dg(\tau')}{d\tau'}\right|_{\tau'=0} = 0, & \left.\frac{dg(\tau')}{d\tau'}\right|_{\tau'=1} = 0, \\ \left.\frac{d^2g(\tau')}{d\tau'^2}\right|_{\tau'=0} = 0, & \left.\frac{d^2g(\tau')}{d\tau'^2}\right|_{\tau'=1} = 0. \end{array} \tag{2}$$

Assuming $g(\tau')$ has the following functional form

$$g(\tau') = a\tau'^5 + b\tau'^4 + c\tau'^3 + d\tau'^2 + e\tau' + f, \tag{3}$$

the parameters satisfying Eq. (2) are

$$\begin{aligned} a &= \frac{6}{[1-\tau'(0)]^3[1+3\tau'(0)+6\tau'(0)^2]}, \\ b &= \frac{15}{[\tau'(0)-1]^3[1+3\tau'(0)+6\tau'(0)^2]}, \\ c &= \frac{10}{[1-\tau'(0)]^3[1+3\tau'(0)+6\tau'(0)^2]}, \\ d &= 0, \quad e = 0, \\ f &= \frac{\tau'(0)^3[10-15\tau'(0)+6\tau'(0)^2]}{[\tau'(0)-1]^3[1+3\tau'(0)+6\tau'(0)^2]}. \end{aligned} \tag{4}$$

References

(1) Plimpton, S. J. *J. Comput. Phys.* **1995**, *117*, 1.

(2) `http://lammps.sandia.gov`.

(3) Yang, Z.-H. *Phys. Chem. Chem. Phys.* **2020**, DOI: 10.1039/d0cp02855j.

(4) Field, M. J. *J. Chem. Theory Comput.* **2017**, *13*, 2342.