

Manual for CP-VASP (version 2)

<https://github.com/yuanyue-liu-group/CP-VASP>

Yuanyue Liu group

Table of contents

1. Introduction.....	2
2. How to obtain the code.....	3
3. Citing the code.....	3
4. Installation.....	4
5. CP-DFT Theory.....	4
6. Flash Solvation Method.....	6
7. CP-VASP Usage Overview.....	7
8. CP-VASP INCAR Setting Details.....	8
9. Relevant VASP INCAR Settings.....	12
10. Interpreting the Results.....	13
11. Troubleshooting.....	14
12. References.....	14

1. Introduction

Thank you for taking interest in CP-VASP^{1,2}! CP-VASP is a patch to the Vienna Ab-Initio Simulation Package (VASP)^{3,4} to enable DFT calculations under constant electrode potential. This is essential for accurately simulating the electrochemical interface, where the Fermi level is controlled by external electrode potential. CP-VASP enables fixed-structure calculation, structural relaxation, and molecular dynamics (MD) simulations under a given electrode potential.

As shown Fig. 1(a), the supercell for CP-VASP is made of an explicit region with atoms of interest and an implicit region that models the solution as a dielectric continuum containing point charges (where the dielectric continuum represents the solvent, and the point charges represent the electrolyte ions). The supercell is connected to an “electrode” with potential U_{ext} . CP-VASP tunes the number of electrons in the explicit region to change the average Fermi level to a target value set by U_{ext} . These net electronic charges are balanced by the ionic charges in the implicit region to keep the system charge neutral.

CP-VASP has various algorithms to tune the electron number. Particularly, for MD simulations, it can emulate the “true” grand-canonical ensemble of electrons to produce the correct Fermi level fluctuation (see Fig. 1(b)). Moreover, it implements “flash solvation” technique which avoids the “flying solvent” problem at the explicit-implicit interface (see section 6). This technique also significantly accelerates the MD simulation.

CP-VASP was developed by Zachary Levell*, Ruoyu Wang*, Saerom Yu*, Xunhua Zhao, and Yuanyue Liu. (* These authors contributed equally)

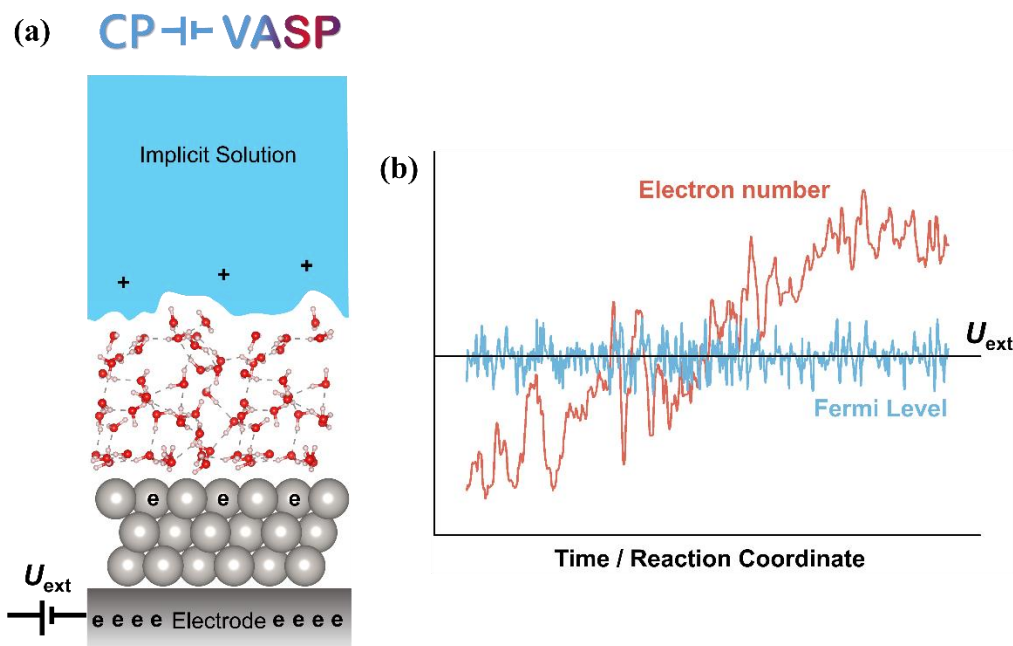


Fig. 1. (a) The CP-VASP model, which features an explicitly defined atomic region and implicit solution. The system is connected to an external potentiostat. (b) Time evolution of the electron number and Fermi level. Whether the system is in equilibrium or undergoing a reaction, the Fermi level fluctuates around the external potential according to the grand-canonical ensemble.

2. How to obtain the code

The code is available upon request at the following link:

<https://docs.google.com/forms/d/e/1FAIpQLSdjnABLY26jora50jedL5X6bj5Pc7BTeys5pmclstlWs46Lw/viewform>

3. Citing the code:

If you use this code in your work, please cite the following papers:

- Zhao, X.; Liu, Y. Origin of Selective Production of Hydrogen Peroxide by Electrochemical Oxygen Reduction. *Journal of the American Chemical Society* **2021**, 143 (25), 9423-9428.

- Yu, S.; Levell, Z.; Jiang, Z.; Zhao, X.; Liu, Y. What Is the Rate-Limiting Step of Oxygen Reduction Reaction on Fe–N–C Catalysts? *Journal of the American Chemical Society* **2023**, 145 (46), 25352-25356.

4. Installation

CP-VASP is intended to be applied to VASP version 6. It requires installing either VASPsol or VASPsol++ before being applied:

VASPsol: <https://github.com/henniggroup/VASPsol/tree/master>

VASPsol++: https://gitlab.com/cplaisance/vaspsol_pp.

CP-VASP is compatible with VASPsol++ for VASP version 6.3 and with VASPsol for VASP versions 6.4. CP-VASP uses the cp-vaspsol++.patch file for VASPsol++ and the cp-vaspsol.patch file for VASPsol, respectively. To apply the patch, navigate to the /src directory in your VASP folder and copy the appropriate patch file into it. Then execute the following command (assuming you are using VASP version 6.3 and VASPsol++):

```
patch -p0 < cp-vaspsol++.patch
```

The VASPsol++ patch file modifies the main.F and reader.F, whereas the VASPsol patch files modify the main.F, pot.F, reader.F, and the solvation.F files. The .objects file does not need to be edited. Once the patch is complete, recompile VASP. In our experience, make veryclean is typically not necessary but is worth trying if compilation errors occur.

5. CP-DFT Theory

In constant potential density functional theory (CP-DFT), the number of electrons (n) is changed to match the Fermi level (referenced to the electrolyte) with a target value, which is set by the electrode potential. For convenience, we define the “interfacial potential” of the system (U) as:

$$E_F - e\Phi = -eU \quad (\text{Eq. 1})$$

where E_F is the Fermi level of the system and Φ is the electrostatic potential in the electrolyte region. CP-DFT controls the system so that:

$$U = U_T \quad (\text{Eq. 2})$$

where U_T is the target potential.

For a given structure, determining the n to satisfy Eq. 2 corresponds to a 1D root-finding problem for the U vs n profile, which can be approached in several ways. CP-VASP currently uses the Secant method, which strictly converges the internal potential to match the external potential so that there are (ideally) no fluctuations in the potential. The Secant method starts from two known datapoints: (n_0, U_0) and (n_1, U_1) . The electron number at step i is calculated as:

$$n_i = n_{i-1} + C (U_T - U_{i-1}) \quad (\text{Eq. 3})$$

where C mimics the capacitance of the system, which is calculated as:

$$C = (n_{i-1} - n_{i-2}) / (U_{i-1} - U_{i-2}) \quad (\text{Eq. 4})$$

For each step, DFT SCF cycle is performed to determine the U for given n . Eqs. 3 and 4 are iterated until the difference between U and U_T becomes sufficiently small.

For structure relaxation, one can run multiple electronic steps (i.e. SCF cycles) for every structure encountered during the relaxation to ensure Eq. 2 is satisfied for each structure (NESHEME = 3). However, since the final structure is typically of greater interest, a more efficient approach is to only run one SCF cycle for each structure (NEScheme = 2). Alternatively, one can run multiple SCF cycles for the first ionic step, and then run only one SCF cycle for each sequential step (NEScheme = 1).

For MD, one can still use the Secant method (NEScheme = 1, 2, 3) to converge the interfacial potential. However, unlike static calculations (e.g. relaxation) where U equals the external electrode potential ($U = U_T$), U should fluctuate around U_T under thermodynamic equilibrium (i.e. $\langle U \rangle = U_T$) during CP-MD. As such, the Secant method does not capture the real dynamics of the potential. Dynamic equations for this system have previously been developed to couple the system to an external potentiostat⁵. The simplest approach is to introduce a fictitious force on the electron number as the difference between the U and U_T (NEScheme = 4). The equations of motion become identical to an ideal spring. In CP-VASP, this method is integrated with time using the Velocity-Verlet algorithm. However, this scheme does not include the effects of temperature. To address this, the potentiostat can be coupled to a thermostat (either the same reservoir as the ionic positions or a separate one). CP-VASP currently allows the potentiostat to be connected to an isolated

thermostat (NESCHEME = 5). The equations of motion for this approach are given below:

$$\dot{n} = \frac{P_n}{M_n} \quad (\text{Eq. 5})$$

$$\dot{P}_n = U - U_T - P_n \frac{P_\xi}{M_\xi} \quad (\text{Eq. 6})$$

$$\dot{\xi} = \frac{P_\xi}{M_\xi} \quad (\text{Eq. 7})$$

$$\dot{P}_\xi = \frac{P_n^2}{M_n} - kT_\xi \quad (\text{Eq. 8})$$

where n is the electron number at the current step, P_n is the fictitious momentum of the electron number at the current step, and M_n is the fictitious mass that controls the coupling between the system and electron bath. ξ is the control variable for the thermostat, and P_ξ , M_ξ , and T_ξ , are the corresponding momentum, mass, and temperature for the thermostat, respectively. These equations are time-integrated using the N ose-Hoover chain method^{6,7}.

6. Flash Solvation Method

Performing MD with implicit solvation can break the connections of explicit solvent molecules and cause them to drift into the implicit region. To address this "flying solvent" problem, CP-VASP uses the "flash solvation" method. In this method, implicit solution is only used to reference the potential, whereas the forces are calculated without any implicit solution. Every few ionic steps, CP-VASP uses the implicit solvation to measure the potential and update the number of electrons accordingly, and then turns off the implicit solvation to calculate the force. This approach helps the solvent molecules stay connected and avoids the "flying solvent" problem.

Another advantage of this technique is that the computational efficiency is improved. By only using implicit solvation once every few steps, the computational cost is reduced. This method is controlled by the VASP INCAR parameter NEADJUST, which is discussed in further detail in section 8.

7. CP-VASP Usage Overview

To perform CP-DFT calculations, add the following tags into INCAR:

LCEP = .TRUE.

NESHEME = [INTEGER]

TARGETMU = [REAL]

Turning on the constant potential calculation (LCEP = .TRUE.) performs CP-DFT calculations to reach the desired electrode potential (TARGETMU) using the electronic charge scheme (NESHEME) specified by the user. Details about these tags are shown later in section 5. We note that CP-VASP is only compatible with IBRION set to -1, 0, 1, 2, and 3.

The user can also add the following optional tags:

FERMICONVERGE = [REAL]

INIT_ECHANGE = [REAL]

ECHANGE_MAX = [REAL]

CAP_MAX = [REAL]

NEADJUST = [INTEGER]

LSOLOFF = [LOGICAL]

Vne = [REAL]

Mne = [REAL]

T_eta = [REAL]

M_eta = [REAL]

eta_length = [INTEGER]

FERMICONVERGE determines the convergence criterion of the electrolyte-referenced Fermi level to the target energy $-eU_T$. INIT_ECHANGE is the magnitude of the initial electron number change. ECHANGE_MAX and CAP_MAX are the maximum magnitude of the electron number change and the capacitance, respectively. NEADJUST controls how often the electron number is adjusted. The remaining parameters are only relevant when using NESHEME = 4 and NEScheme = 5. These tags determine the behavior the potentiostat when using the dynamic equations of motion. Further details for each of these parameters are given below.

8. CP-VASP INCAR Setting Details

LCEP = [LOGICAL]

Default: LCEP = .FALSE.

Setting LCEP = .TRUE. will turn on the CP for DFT calculations. Regular VASP calculations can still be performed by setting LCEP = .FALSE.

NESHEME = [INTEGER]

Default: NESHEME = 1

NESHEME determines which electron adjustment scheme is used.

NESHEME = 1 (Hybrid-scheme)

The electron number is tuned via the Secant method. For the first ionic step, Eq. 3 is repeated over multiple SCF cycles until the electrolyte-referenced Fermi level is sufficiently converged. The rest of the ionic steps use a single SCF cycle to update the number of electrons.

NESHEME = 2 (Single-SCF scheme)

Using the Secant method, each ionic step has a single electron number adjustment.

NESHEME = 3 (Multi-SCF scheme)

The electrolyte-referenced Fermi level is converged at each ionic step via the Secant method, performing as many electron number adjustments as necessary.

NESHEME = 4 (Dynamic “constant-energy” scheme)

This scheme uses the first scheme proposed by Otani et al⁵. This method evolves the electron number dynamically so that the force is akin to a fictitious spring (see Eqs. 5 and 6). The equations are integrated using the Velocity Verlet algorithm.

NESHEME = 5 (Dynamic “constant-temperature” scheme)

This scheme uses the third scheme proposed by Otani et al⁵. This method extends NESHEME = 4 by coupling the potentiostat to a thermostat (separate from the ionic positions). The equations are integrated using the N ose-Hoover Chain method.

TARGETMU = [REAL]

Default: TARGETMU = -4.6

TARGETMU sets the targeted Fermi level referenced to the electrolyte $-eU_T$. It is useful to reference the potential to the SHE to enable comparison with experiments, but correctly referencing the potential is challenging. The reference potential can be determined in a few ways. If experimental data is available, we recommend finding the experimental difference between the SHE potential and the potential of zero charge ($U_{\text{SHE}} - U_{\text{PZC}}$). Next, calculate the potential of zero charge computationally using the implicit solvation model (for VASPsol++ this is simply E-fermi; for VASPsol this is E-fermi + FERMI_SHIFT). Then set TARGETMU according to the observed U_{PZC} plus this difference to obtain the desired potential. If experimental data is unavailable, use this method for similar materials with experimental data and use the average difference. Other processes are described in greater detail in our review paper⁸. Note that the value of Fermi level referenced to the electrolyte for the SHE can change with the functional and solvation model and should be re-calculated when working with a new functional/solvation model. We recommend ensuring that TARGETMU is not within a band gap of the system, as there are no electron numbers that produce the desired potential and CP-VASP will be unable to converge the interfacial potential.

FERMICONVERGE = [REAL]

Default: FERMICONVERGE = 0.05

FERMICONVERGE sets the convergence criterion for the electrolyte-referenced Fermi level. When the difference between it and TARGETMU is smaller than FERMICONVERGE, the potential is considered to have reached the target value.

INIT_EXCHANGE = [REAL]

Default: INIT_EXCHANGE= 0.1

This parameter sets the magnitude of the first electron number change. The code automatically determines whether the electron number should decrease or increase

based on the electrolyte-referenced Fermi level relative to the target energy. INIT_ECHANGE is only applicable to NEScheme = [1,2,3].

ECHANGE_MAX = [REAL]

Default: ECHANGE_MAX = 0.0

This parameter determines the maximum magnitude of the electron number change. In most cases, this parameter is not necessary and is turned off by default (by setting ECHANGE_MAX = 0.0). Large changes in the electron number may cause the SCF cycles to fail to converge. Setting ECHANGE_MAX to a positive value can prevent the SCF cycles from failing, but also reduce the speed of convergence. ECHANGE_MAX is only applicable to NEScheme = [1,2,3].

CAP_MAX = [REAL]

Default: CAP_MAX = 2.0

This parameter determines the maximum capacitance when NEScheme = [1,2,3]. Increasing CAP_MAX will allow the electron number to change by a greater amount in a single electron adjustment step. Limiting the capacitance prevents CP-VASP from greatly overshooting the target potential at the expense of some computational speed.

NEADJUST = [INTEGER]

Default: NEADJUST = 1

This parameter controls how often the number of electrons is adjusted and is mostly relevant during MD simulations. We have previously observed that it is sufficiently accurate to only perform an electron adjustment every few ionic steps^{1, 2}. The advantage of applying this tag is the computational efficiency without considerably disturbing the local structure and reaction near the surface and preventing the “flying solvent” problem. When the explicit solvation is present, the local structure won’t be significantly affected since it is far from the implicit solvation region. To maximize efficiency, CP-VASP only turns on implicit solvation every NEADJUST steps, calculates the interfacial potential, and then turns off

implicit solvation and proceeds with MD. Note that the electron adjustment step is always performed on the first ionic step, and then occurs every NEADJUST steps.

LSOLOFF = [LOGICAL]

Default: LSOLOFF = .TRUE. (if IBRION = 0 and LCEP = .TRUE.)

= .FALSE. (else)

This parameter turns off the implicit solution by running an additional SCF calculation in vacuum for every ionic step where an electron is adjusted. Thus, the implicit solution is only used to reference the Fermi level, whereas the forces are calculated without solvation.

Vne = [REAL]

Default: Vne = 0.0

When using NEScheme = 4 and NEScheme = 5, this parameter determines the initial momentum of the potentiostat. This parameter is mainly useful when running a continuation job.

Mne = [REAL]

Default: Mne = 660.74

When using NEScheme = 4 and NEScheme = 5, this parameter determines the fictitious “mass” of the potentiostat; if Mne is increased, the potentiostat is more resistant to changing the electron number, thereby causing the fluctuations to have a longer period. The optimal value can be obtained from $M_{ne} = \frac{1}{(C_{ne}) * (\omega_{ne})^2}$, where C_{ne} is the size-dependent capacitance and ω_{ne} is the frequency of the potentiostat fluctuations. The capacitance is defined as $C_{ne} = \frac{dn_e}{d\phi}$ and scales with the size of the system. The default value is calculated assuming the frequency is 3700 cm⁻¹ (which is the O-H bond vibrational frequency) and the capacitance is 0.123 eV⁻¹ (which we obtained from a Pt unit cell). The capacitance can be estimated using CP-VASP by running a fixed structure calculation with the initial Fermi level reasonably distance

from the target potential, where $C_{ne} \approx \frac{\Delta n_e}{\Delta \phi}$. The mass must be given in units of $\text{eV} \cdot \text{fs}^2$ for reasonable results.

T_eta = [REAL]

Default: T_eta = TEBEG

When using NESHEME = 5, this parameter sets the temperature of the thermostat that is coupled to the potentiostat. The default value is set to the temperature of the ionic position thermostat but can be adjusted; larger temperatures correspond to greater fluctuation amplitudes. The value should be given in units of K.

M_eta = [REAL]

Default: M_eta = $k_b * T_{eta} * 81.27$

When using NESHEME = 5, this value is the fictitious mass of the thermostat that is coupled to the potentiostat. The default value is chosen so that the frequency of fluctuations for the thermostat is 3700 cm^{-1} . The value is expressed in units of $\text{eV} \cdot \text{fs}^2$.

eta_length = [INTEGER]

Default: eta_length = 1

When using NESHEME = 5, this parameter is the chain length of the N se-Hoover Chain thermostat that is coupled to the potentiostat.

9. Relevant VASP INCAR Settings

IBRION = [-1, 0, 1, 2, 3]

CP-VASP is not currently compatible with all IBRION settings; the values listed above are implemented into CP-VASP. Some details for each of these settings are noted below:

IBRION = -1: This corresponds to a fixed-structure calculation.

IBRION = 0: This runs ab initio molecular dynamics (AIMD) with DFT-calculated forces.

IBRION = 1, 2, 3: These values perform ionic-relaxation calculations.

NELECT = [REAL]

If you know the charge of the system at the target potential, it is helpful to initialize the electron number to reduce the computational cost. This parameter and ICHARG are particularly useful for continuation jobs.

LSOL = [LOGICAL]

CP-VASP currently requires that VASPsol or VASPsol++ is tuned on to correctly reference the Fermi level. Additional inputs are used to describe the solution properties for each model, see the corresponding manuals for more information.

10. Interpreting the Results

CP-VASP includes two additional output lines to the standard VASP output and OSZICAR files:

“CPM-ion: ...”: This line is printed directly after the traditional VASP output line that gives the system energy at the end of each ionic step. The values on this line are as follows: the ionic step number, the total number of electrons, the electrolyte-referenced Fermi level ($E_F - e\Phi$), the target energy ($-eU_T$), and the grand canonical energy (GCE). The code will also print out the instantaneous capacitance (if NEScheme = [1,2,3]) or the velocity of the potentiostat (Vne) (if NEScheme = [4,5]). The GCE calculated by the formula below:

$$\text{GCE} = E_0 + (\text{CNNE} - n) * (-eU_T) \quad (\text{Eq. 7})$$

where E_0 is the standard VASP output energy, CNNE is the charge-neutral electron number, n is the electron number, and U_T is the external electrode potential. The GCE is given in terms of eV. Note that E_0 does *not* account for the change in electron number and is no longer an accurate representation of the system’s total energy.

“CPM-scf: ...”: This line occurs at the end of each SCF cycle when the electrolyte-referenced Fermi level has not yet converged and NESHEME = 3 or (NSTEP = 1 and NESHEME = 1). The values on this line are as follows: the ionic step number, the number of SCF cycles performed this ionic step, the total number of electrons, the electrolyte-referenced Fermi level, the target energy, and the capacitance. The main purpose of this line is to evaluate how quickly the interfacial potential is converging to the target potential. If the target potential cannot be reached for whatever reason, CP-VASP will endlessly attempt to change the electron number and the simulation will not finish. This may happen if your material is an electronic insulator at the chosen potential, or if the density of states is particularly sparse in that region. In this circumstance, try increasing the number of k-points or the orbital energy smearing width (SIGMA).

It is particularly useful to plot the ionic step vs electron number, the ionic step vs electrolyte-referenced Fermi level, the electron number vs electrolyte-referenced Fermi level, and the ionic step vs grand canonical energy. Examples showing the analysis of CP-VASP simulations are shown in a separate document.

11. Troubleshooting

If you run into any errors compiling the VASP code with CP-VASP or the simulations do not run properly, please make a post in the Issues tab on our GitHub page (<https://github.com/yuanyue-liu-group/CP-VASP/issues>). CP-VASP is under development, and we appreciate your feedback.

12. References

- (1) Zhao, X.; Liu, Y. Origin of selective production of hydrogen peroxide by electrochemical oxygen reduction. *Journal of the American Chemical Society* **2021**, *143* (25), 9423-9428.
- (2) Yu, S.; Levell, Z.; Jiang, Z.; Zhao, X.; Liu, Y. What Is the Rate-Limiting Step of Oxygen Reduction Reaction on Fe–N–C Catalysts? *Journal of the American Chemical Society* **2023**, *145* (46), 25352-25356. DOI: 10.1021/jacs.3c09193.
- (3) Kresse, G.; Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science* **1996**, *6* (1), 15-50. DOI: [https://doi.org/10.1016/0927-0256\(96\)00008-0](https://doi.org/10.1016/0927-0256(96)00008-0).
- (4) Kresse, G.; Hafner, J. Ab initio molecular dynamics for liquid metals. *Physical Review B* **1993**, *47* (1), 558-561. DOI: 10.1103/PhysRevB.47.558.
- (5) Bonnet, N.; Morishita, T.; Sugino, O.; Otani, M. First-Principles Molecular Dynamics at a Constant Electrode Potential. *Physical Review Letters* **2012**, *109* (26), 266101. DOI: 10.1103/PhysRevLett.109.266101.

- (6) Martyna, G. J.; Klein, M. L.; Tuckerman, M. Nosé–Hoover chains: The canonical ensemble via continuous dynamics. *The Journal of Chemical Physics* **1992**, *97* (4), 2635-2643. DOI: 10.1063/1.463940 (accessed 9/16/2024).
- (7) Martyna, G. J.; Tuckerman, M. E.; Tobias, D. J.; Klein, M. L. Explicit reversible integrators for extended systems dynamics. *Molecular Physics* **1996**, *87* (5), 1117-1157. DOI: 10.1080/00268979600100761.
- (8) Levell, Z.; Le, J.; Yu, S.; Wang, R.; Ethirajan, S.; Rana, R.; Kulkarni, A.; Resasco, J.; Lu, D.; Cheng, J.; et al. Emerging Atomistic Modeling Methods for Heterogeneous Electrocatalysis. *Chemical Reviews* **2024**, *124* (14), 8620-8656. DOI: 10.1021/acs.chemrev.3c00735.
- (9) Mathew, K.; Kolluru, V. S. C.; Mula, S.; Steinmann, S. N.; Hennig, R. G. Implicit self-consistent electrolyte model in plane-wave density-functional theory. *The Journal of Chemical Physics* **2019**, *151* (23), 234101. DOI: 10.1063/1.5132354 (accessed 9/1/2023).
- (10) Islam, S.; Khezeli, F.; Ringe, S.; Plaisance, C. An implicit electrolyte model for plane wave density functional theory exhibiting nonlinear response and a nonlocal cavity definition. *The Journal of Chemical Physics* **2023**, *159* (23).