Non-adiabatic Molecular Dynamics using Hefei-NAMD

Qijing Zheng

zqj@ustc.edu.cn

Department of Physics
University of Science & Technology of China
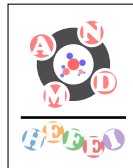
Jul 14, 2022

Hefei-NAMD Logo[1]

- **H**efei **N**on**A**diabatic **M**olecular **D**ynamics (Hefei-NAMD) is a suite of codes that can perform nonadiabatic molecular dynamics based on TSH and CPA on *extended systems*.
  - Single-particle dynamics: charge transfer, *e-h* recombination...
    [*J. Am. Chem. Soc.*, **138**, 13740 (2016)], [*Nano Lett.*, **17**, 6435–6442 (2017)],
  - Spin dynamics including spin-orbit coupling
    [*Phys. Rev. B*, **105**, 085142 (2022)]
  - Exciton dynamics base on *GW*+rtBSE
    [*Sci. Adv.*, **7**, eabf3759 (2021)]
  - Nuclear quantum effects on charge transfer
    [*Sci. Adv.*, **8**, eabo2675 (2022)]

- Interfaces to existing DFT code: `VASP, FHI-aims, Quantum Expresso`...

---

[1]Generated by this script: https://github.com/QijingZheng/Hefei-NAMD/blob/master/logo/genlogo.py

Prof. Jin Zhao

Dr. Qijing Zheng

Dr. Weibin Chu

Dr. Xiang Jiang

Dr. Yongliang Shi

Dr. Chuanyu Zhao

Mr. Zhenfa Zheng

Mr. Aolei Wang

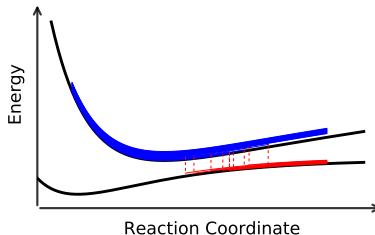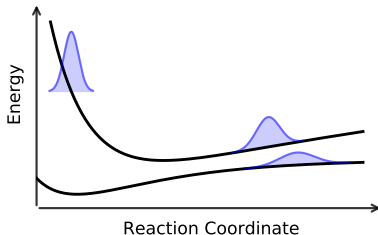and more. . .

1 Hefei-NAMD Intro
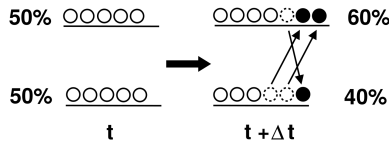
2 A Bit of Theory

3 Learn by Example

Basic Idea

- Electronic wavefunction evolves according to time-dependent Schrödinger equation.
    - $\Phi(\mathbf{r}, \mathbf{R}, t) = \sum_j C_j(t)\psi_j(\mathbf{r}, \mathbf{R})$ $\Rightarrow$ $i\hbar \dot{C}_j(t) = \sum_k C_k(t)\left[H_{jk} - i\hbar\dot{\mathbf{R}} \cdot \mathbf{d}_{jk}\right]$

    - $H_{jk} = \langle\psi_j|\hat{\mathcal{H}}_{el}|\psi_k\rangle$

    - Non-adiabatic Couplings (NAC): $\mathbf{d}_{jk} = \langle\psi_j|\nabla_{\mathbf{R}}|\psi_k\rangle$
- An ensemble of *independent* nuclear trajectories is considered.
- Each trajectory propagates on *one single* potential energy surface at any given time.
    - $M_\alpha \ddot{\mathbf{R}}_\alpha = -\nabla_\alpha E_k^{el}(\mathbf{R})$
- Hops of trajectories between electronic states is possible.
    - Tully's Fewest-Switches algorithm et al.

fewest switches

50% OOOOO → OOOOO● 60%

50% OOOOO → OOOO○ 40%

t   t + Δt

50% OOOOO → OOOO●● 60%

50% OOOOO → OOO○● 40%

t   t + Δt
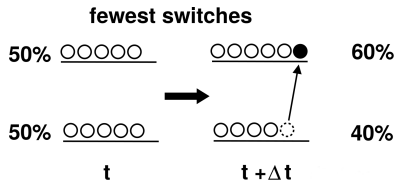
Assumptions

- Ensemble of independent trajectories have same coefficients $C_j(t)$.

- Internal consistency condition $N_j(t) \propto C_j^*(t) C_j(t) = \rho_{jj}(t)$.

- Hops from j to different $k \neq j$ are independent.

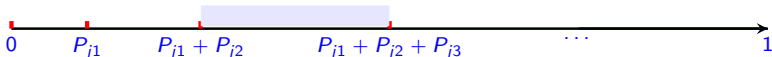- Overall trajectory hops should be minimum.

- Hopping Probability

  Transition from current state j to state $k \neq j$ is allowed only if population of state j is *decreasing*.

  $$P_{jk}(t, \Delta t) = \max\left(-\frac{2 \int_{t}^{t+\Delta t} dt \left[\hbar^{-1} \operatorname{Im}(\rho_{jk} H_{jk}) - \operatorname{Re}(\rho_{jk} \mathbf{d}_{jk} \cdot \dot{\mathbf{R}}) \right]}{\rho_{jj}}, 0 \right)$$

- Which state to hop

  $$\sum_{l=1, l \neq j}^{k-1} P_{jl} < \xi < \sum_{l=1, l \neq j}^{k} P_{jl} \quad \text{then} \quad j \to k$$

  0   P_{j1} $P_{j1} + P_{j2}$ $P_{j1} + P_{j2} + P_{j3}$ $\cdots$ 1

- After trajectory hops
  - Energy should be conserved after the hop e.g. by rescaling the nuclear velocity in the direction of non-adiabatic coupling.
    - $\mathbf{E}_j + \mathbf{K}_j = \mathbf{E}_k + \mathbf{K}_k \quad \Rightarrow \quad \mathbf{R}_{\alpha}^{k} = \mathbf{R}_{\alpha}^{j} - M_{\alpha}^{-1} \lambda \mathbf{d}_{jk}$
  - Hops *rejected* when energy conservation *not satisfied*.

1 Hefei-NAMD Intro

2 A Bit of Theory
   - Trajectory Surface Hopping
   - Numerical Evaluation of NAC
   - Spin-orbit Coupling

3 Learn by Example

The key quantity in NAMD is the non-adiabatic couplings (NACs)

$$\mathbf{D}_{jk} = \langle \psi_j | \nabla_{\mathbf{R}} | \psi_k \rangle \cdot \dot{\mathbf{R}} \tag{1}$$

$$= \langle \psi_j | \frac{\partial}{\partial t} | \psi_k \rangle \approx \frac{\langle \psi_j(t) | \psi_k(t + \delta t) \rangle - \langle \psi_j(t + \delta t) | \psi_k(t) \rangle}{2 \delta t} \tag{2}$$

where $\psi_j(t)$ and $\psi_j(t + \delta t)$ are the j-th Kohn-Sham (KS) wavefunctions (WFCs) at adjacent time steps. The KS WFCs can be expanded in a basis set $\{\chi_n(\mathbf{r})\}$, i.e. $\psi_k(\mathbf{r}) = \sum_{n=1}^{N_b} c_{kn} \cdot \chi_n(\mathbf{r})$, where c_{kn} is the expanding coefficients.

- From Eq. (1), one can show that [2]

$$D_{jk} = \dot{\mathbf{R}} \cdot \langle \psi_j | \nabla_{\mathbf{R}} | \psi_k \rangle = \dot{\mathbf{R}} \cdot \frac{\langle \psi_j | \nabla_{\mathbf{R}} \mathcal{H} | \psi_k \rangle}{\varepsilon_k - \varepsilon_j} \tag{3}$$

$$= \frac{\dot{\mathbf{R}}}{\varepsilon_k - \varepsilon_j} \cdot \sum_{n,m} c_{jn}^* c_{km} \left[\frac{H_{nm}}{\partial \mathbf{R}} - \varepsilon_j \left\langle \frac{\partial \chi_n}{\partial \mathbf{R}} \Big| \chi_m \right\rangle - \varepsilon_k \left\langle \chi_n \Big| \frac{\partial \chi_m}{\partial \mathbf{R}} \right\rangle \right] \tag{4}$$

- From Eq. (2), we have

$$D_{jk} \approx \frac{\sum_{n,m} \left[c_{jn}^*(t) \langle \chi_n(t) | \chi_m(t + \delta t) \rangle c_{km}(t + \delta t) - c_{jn}^*(t + \delta t) \langle \chi_n(t + \delta t) | \chi_m(t) \rangle c_{km}(t) \right]}{2 \delta t} \tag{5}$$

- The blue terms in Eq. (4) are *zero for plane-waves* and easy to calculate for atomic basis. However, the red terms may require a DFPT calculation, which might be time-consuming for large systems.

- The orange terms in Eq. (5) are two-center integrals, which involve basis functions at adjacent time-steps and are *always 1 for plane-waves*.

  - They might be related to the blue terms in Eq. (4) by

$$\langle \chi_n(t) | \chi_m(t + \delta t) \rangle \approx \langle \chi_n(t) | \chi_m(t) \rangle + \delta t \cdot \dot{\mathbf{R}} \cdot \left\langle \chi_n \left| \frac{\partial \chi_m}{\partial \mathbf{R}} \right. \right\rangle \tag{6}$$

  - For atomic basis on radial logarithmic grid, can be done in reciprocal space utilizing *fast Hankel transform*. [3]

    - Suppose the basis function can be written as mulitiplication of radial and angular part, then [4]

$$\chi_j^{\ell m}(\mathbf{r}; \mathbf{R}) = f_j(|\mathbf{r} - \mathbf{R}|) \cdot Y_\ell^m(\theta, \phi) \quad \overset{\mathrm{FT}}{\longleftrightarrow} \quad \mathcal{F}[\chi](\mathbf{G}) = i^\ell \cdot h_j(G) Y_\ell^m(\hat{\mathbf{G}}) \cdot e^{i \mathbf{G} \cdot \mathbf{R}} \tag{7}$$

      where $h_j(G)$ is the Hankel transform of $f_j(r)$.

    - The orange terms then reduce to (change index from n, m to j, k to avoid ambiguity)

$$\langle \chi_j^{\ell m}(t) | \chi_k^{\ell' m'}(t + \delta t) \rangle = \sum_{\mathbf{G}} i^{(\ell' - \ell)} \cdot h_j(G)^* Y_\ell^{m*}(\hat{\mathbf{G}}) \cdot h_k(G) Y_{l'}^{m'}(\hat{\mathbf{G}}) \cdot e^{i \mathbf{G} \cdot [\mathbf{R}(t + \delta t) - \mathbf{R}(t)]} \tag{8}$$

      The magenta terms are time-independent and can be calculated once for all.

---

[2] *J. Chem. Phys.*, **138**, 154106 (2013)

[3] http://staff.ustc.edu.cn/~zqj/posts/FourierTransform-Radial-Function/

[4] The phase term $e^{i \mathbf{G} \cdot \mathbf{R}}$ is due to shifting property of FT.

In plane-wave (PW) basis code, e.g. `VASP`

$$\chi_n(\mathbf{r}) = e^{i\,\mathbf{G}_n\cdot\mathbf{r}} \tag{9}$$

where the plane-waves vectors are integer multiples of the reciprocal basis vectors

$$\mathbf{G}_n = \sum_{j=1}^{3} n_j \cdot \mathbf{b}_j \quad (n_j = 0, 1, 2, 3\ldots) \tag{10}$$

- In PW codes, *pseudopotentials* are often used, hence the WFC is a *pseudo-WFC*, often denoted as $\tilde{\psi}_k(\mathbf{r})$.
- The WFCs, charge density etc. are represented on an *uniform* 3D grid.
- As long as the cell shape and the energy cutoff are fixed, the basis set is determined.
- The PWs are *delocalized non-centered orthogonal* functions. The overlap between KS WFCs can be simply written as

$$\langle\psi_j(t)|\psi_k(t+\delta t)\rangle = \sum_n c_{jn}^*(t) \cdot c_{kn}(t+\delta t) \tag{11}$$

- In PAW formalism, all-electron (AE) correction to the NAC can be obtained. [5]

---

[5] *J. Phys. Chem. Lett.*, **11**, 23, 10073 (2020)
https://github.com/WeibinChu/CA-NAC

In atomic-orbitals basis code, e.g. in `FHI-aims`

$$\chi_n(\mathbf{r}; \mathbf{r}_0) = \frac{u_n(r)}{r} \cdot Y_{lm}(\theta, \phi) \tag{12}$$

where $u_n(r)$ is the radial function centered at $\mathbf{r}_0$ and Y_{lm} is the real-value spherical harmonics. [6]

- In FHI-aims, the KS WFCs are by definition *all-electron* WFCs.
- The basis function is *atom-centered non-orthogonal* function often represented on a non-uniform grid, e.g. logarithmic radial grid, which means that the basis implicitly *depends on the atomic positions*.
- Assuming that positions of the nuclei does NOT change much between adjacent time steps [7]

$$\langle \chi_n(t) | \chi_m(t + \delta t) \rangle \approx \langle \chi_n(t) | \chi_m(t) \rangle = S_{nm}(t) \tag{13}$$

Eq. (13) is the so-called *concentric approximation*. Under this approximation, the overlap of KS WFCs can be written as

$$\langle \psi_j(t) | \psi_k(t + \delta t) \rangle \approx \sum_{n,m} c_{jn}^*(t) \cdot S_{nm}(t) \cdot c_{km}(t + \delta t) \tag{14}$$

- The final form for NAC

$$\mathbf{D}_{jk} = \frac{\sum_{n,m} c_{jn}^*(t) \cdot S_{nm}(t) \cdot c_{km}(t + \delta t) - \sum_{n,m} c_{jn}^*(t + \delta t) \cdot S_{nm}(t) \cdot c_{km}(t)}{2\delta t} \tag{15}$$

---

[6] http://staff.ustc.edu.cn/~zqj/posts/Plotly-Spherical-Harmonics
[7] For hydrogen atom under room temperature, the velocity is about 0.016 Å/fs

- *Arbitrary* phase can be assigned to the wavefunction $\psi_j(t)$ without affecting the *physical observable quantities*.

- Nondiagonal matrix elements $\langle\psi_j|\hat{S}|\psi_k\rangle$ with $j \neq k$, where ψ_j and ψ_k are of *same* or *different* times, may be *discontinuous* due to the arbitrary phase that *do NOT* cancel out.

- Phase correction in NAC from finite-difference: align the phase to previous time step.

  1. evaluate the phase difference

  $$e^{i\theta_{jk}(t+\Delta t)} = \frac{\langle\psi_j(t)|\psi_k(t+\Delta t)\rangle}{\|\psi_j(t)\| \cdot \|\psi_k(t+\Delta t)\|}$$

  2. reverse the phase difference and apply to the nondiagonal matrix element

  $$S_{jk}(t+\Delta t) = e^{-i\theta_{jk}(t+\Delta t)} \cdot \langle\psi_j(t+\Delta t)|\hat{S}|\psi_k(t+\Delta t)\rangle$$
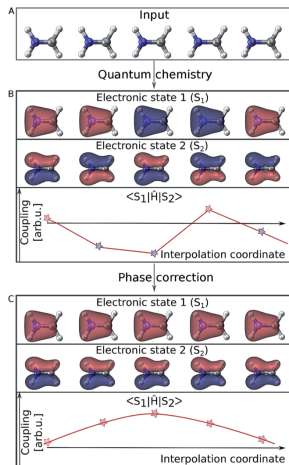

Figure – Effect of phase correction.[8]

[8] *Chem. Sci.*, **10**, 8100 (2019); *J. Chem. Phys.*, **122**, 034105 (2005); *J. Phys. Chem. Lett.*, **9**, 6096 (2018)
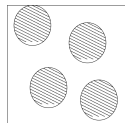
Spin-orbit Hamiltonian and AE wavefunctions in PAW formalism

AE

$$\hat{\mathcal{H}}^{soc}(\mathbf{r}) = \frac{\hbar \vec{\sigma} \cdot \vec{\mathbf{p}} \times \nabla v_{KS}(\mathbf{r})}{4m_e^2 c^2};$$

Due to the derivatives, SOC is dominated in core region.

$$|\psi_{n\sigma}\rangle = \sum_{i=\{\ell,m\},\alpha} \langle \tilde{p}_{i\sigma}|\tilde{\psi}_{n\sigma}\rangle |\phi_{i\sigma}^{\alpha}\rangle$$

AE in PAW sphere

we have

$$\langle \psi_{n\sigma_1}|\hat{\mathcal{H}}^{soc}|\psi_{m\sigma_2}\rangle = \langle \tilde{\psi}_{n\sigma_1}| \sum_{\alpha,i,j} \langle \phi_{i\sigma_1}^{\alpha}|\hat{\mathcal{H}}^{soc}|\phi_{j\sigma_2}^{\alpha}\rangle |\tilde{p}_{i\sigma_1}^{\alpha}\rangle\langle \tilde{p}_{j\sigma_2}^{\alpha}|\tilde{\psi}_{m\sigma_2}\rangle \tag{16}$$

$$\langle \phi_{i\sigma_1}^{\alpha}|\hat{\mathcal{H}}^{soc}|\phi_{j\sigma_2}^{\alpha}\rangle = -\frac{1}{2m_e^2 c^2}\langle Y_i^{\alpha};\sigma_1|\hat{\mathbf{S}} \cdot \hat{\mathbf{L}}|Y_j^{\alpha};\sigma_2\rangle\langle f_i^{\alpha}|\frac{1}{r}\frac{dv_{KS}^{\alpha}}{dr}|f_j^{\alpha}\rangle \tag{17}$$

assuming spherical symmetry

real spherical harmonic

radial function

Two-step Procedure

❶ SCF Calculation

$$\hat{\mathcal{H}}^0|\psi_{n\sigma}\rangle = \varepsilon_{n\sigma}|\psi_{n\sigma}\rangle$$

❷ Diagonalization of $H_{n\sigma_1,m\sigma_2}$

$$H_{n\sigma_1,m\sigma_2} = \varepsilon_{n\sigma_1}\delta_{n\sigma_1,m\sigma_2} + \langle \psi_{n\sigma_1}|\hat{\mathcal{H}}^{soc}|\psi_{m\sigma_2}\rangle$$

The SOC matrix elements

$$\langle \psi_{n\sigma_1} | \hat{\mathcal{H}}^{soc} | \psi_{m\sigma_2} \rangle = \sum_{\alpha, i, j} \langle \tilde{\psi}_{n\sigma_1} | \tilde{p}^{\alpha}_{i\sigma_1} \rangle \langle \phi^{\alpha}_{i\sigma_1} | \hat{\mathcal{H}}^{soc} | \phi^{\alpha}_{j\sigma_2} \rangle \langle \tilde{p}^{\alpha}_{j\sigma_2} | \tilde{\psi}_{m\sigma_2} \rangle$$

- The projector coefficients $\langle \tilde{p}_i | \tilde{\psi}_n \rangle$ is contained in `NormalCAR`, which can be easily output and parsed.

$$\langle \phi^{\alpha}_{i\sigma_1} | \hat{\mathcal{H}}^{soc} | \phi^{\alpha}_{j\sigma_2} \rangle = -\frac{1}{2m_e^2 c^2} \langle Y_i^{\alpha}; \sigma_1 | \hat{\mathbf{S}} \cdot \hat{\mathbf{L}} | Y_j^{\alpha}; \sigma_2 \rangle \langle f_i^{\alpha} | \frac{1}{r} \frac{dv_{KS}^{\alpha}}{dr} | f_j^{\alpha} \rangle$$

- This term is *not as easily obtained*, where f_i^{α} is the radial function, Y_i^{α} is the real spherical harmonics and KS potential $v_{KS}^{\alpha} = v_{ext}^{\alpha} + v_{xc}^{\alpha} + v_{Har}^{\alpha}$ depend on the charge density of each atom site.

$$\langle Y_i^{\alpha}; \sigma_1 | \hat{\mathbf{S}} \cdot \hat{\mathbf{L}} | Y_j^{\alpha}; \sigma_2 \rangle$$

  - Only $\ell \in (0, 3]$ and $\ell_i = \ell_j$ are considered in the evaluation of this term in VASP.
  - VASP first evaluates it in the basis of complex spherical harmonics, then transforms into the real special harmonics basis.

$$\langle f_i^{\alpha} | \frac{1}{r} \frac{dv_{KS}^{\alpha}}{dr} | f_j^{\alpha} \rangle$$

  - For each angular quantum number ℓ, there are $2\ell + 1$ spherical harmonics, the corresponding radial function is the same for all these harmonics.
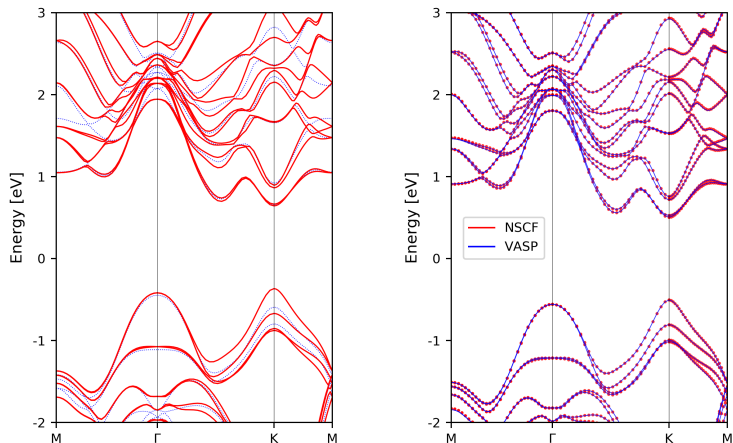
Figure – Left panel: comparison between SOC (red solid) and normal (blue dashed) band structure. Right panel: comparison of the band structure obtained with present method and VASP.[9]

---

[9]https://github.com/QijingZheng/VaspBandUnfolding/blob/master/spinorb.py

The time-dependent Schrödinger equation

$$i\hbar\frac{\partial|\Psi(\mathbf{r},\mathbf{R}(t),\mathbf{s},t)\rangle}{\partial t} = \hat{\mathcal{H}}^{tot}(\mathbf{r},\mathbf{R}(t),\mathbf{s})|\Psi(\mathbf{r},\mathbf{R}(t),\mathbf{s},t)\rangle \tag{18}$$

where the total Hamiltonian is given by

$$\hat{\mathcal{H}}^{tot}(\mathbf{r},\mathbf{R}(t),\mathbf{s}) = \hat{\mathcal{H}}^{0}(\mathbf{r},\mathbf{R}(t)) + \hat{\mathcal{H}}^{soc}(\mathbf{r},\mathbf{R}(t),\mathbf{s}) \tag{19}$$

by expanding the wavefunction with a *basis set* $\{|\psi_i\rangle\}$ or *representation*

$$|\Psi\rangle = \sum_i |\psi_i\rangle\langle\psi_i|\Psi\rangle = \sum_i c_i|\psi_i\rangle \tag{20}$$

and substituting eq (20) into eq (18), we have

$$\frac{\partial c_j(t)}{\partial t} = -\sum_i \left[i\hbar^{-1}\langle\psi_j|\hat{\mathcal{H}}^{tot}|\psi_i\rangle + \langle\psi_j|\frac{\mathrm{d}}{\mathrm{d}t}|\psi_i\rangle \right] c_i(t)$$

$$= -\sum_i \left[i\hbar^{-1}\langle\psi_j|\hat{\mathcal{H}}^{0}|\psi_i\rangle + i\hbar^{-1}\langle\psi_j|\hat{\mathcal{H}}^{soc}|\psi_i\rangle + \langle\psi_j|\frac{\mathrm{d}}{\mathrm{d}t}|\psi_i\rangle \right] c_i(t)$$

$$= -\sum_i \left(i\hbar^{-1}H_{ji}^{0} + i\hbar^{-1}H_{ji}^{soc} + T_{ji} \right) c_i(t) \tag{21}$$

Expand the wavefunction in the "*spin-diabatic* representation"

$$|\Phi\rangle = \sum_{n\sigma} c_n^\sigma |\psi_n^\sigma\rangle = \sum_n \left[c_n^\uparrow \begin{pmatrix} |\psi_n^\uparrow\rangle \\ 0 \end{pmatrix} + c_n^\downarrow \begin{pmatrix} 0 \\ |\psi_n^\downarrow\rangle \end{pmatrix} \right]; \qquad \hat{\mathcal{H}}^0 |\psi_n^\sigma\rangle = \varepsilon_n^\sigma |\psi_n^\sigma\rangle \qquad (22)$$

and substitute eq. (22) into TDKS, we have

$$\frac{\partial}{\partial t}\begin{pmatrix} c_1^\uparrow \\ c_2^\uparrow \\ \vdots \\ c_n^\uparrow \\ c_1^\downarrow \\ c_2^\downarrow \\ \vdots \\ c_n^\downarrow \end{pmatrix} = \frac{-i}{\hbar} \left(\begin{array}{cccc|cccc} \varepsilon_1^\uparrow + S_{11}^{\uparrow\uparrow} & T_{12}^{\uparrow\uparrow} + S_{12}^{\uparrow\uparrow} & \dots & T_{1n}^{\uparrow\uparrow} + S_{1n}^{\uparrow\uparrow} & S_{11}^{\uparrow\downarrow} & S_{12}^{\uparrow\downarrow} & \dots & S_{1n}^{\uparrow\downarrow} \\ T_{21}^{\uparrow\uparrow} + S_{21}^{\uparrow\uparrow} & \varepsilon_2^\uparrow + S_{22}^{\uparrow\uparrow} & \dots & T_{2n}^{\uparrow\uparrow} + S_{2n}^{\uparrow\uparrow} & S_{12}^{\uparrow\downarrow} & S_{22}^{\uparrow\downarrow} & \dots & S_{2n}^{\uparrow\downarrow} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ T_{n1}^{\uparrow\uparrow} + S_{n1}^{\uparrow\uparrow} & T_{n2}^{\uparrow\uparrow} + S_{n2}^{\uparrow\uparrow} & \dots & \varepsilon_n^\uparrow + S_{nn}^{\uparrow\uparrow} & S_{n1}^{\uparrow\downarrow} & S_{n2}^{\uparrow\downarrow} & \dots & S_{nn}^{\uparrow\downarrow} \\ \hline S_{11}^{\downarrow\uparrow} & S_{12}^{\downarrow\uparrow} & \dots & S_{1n}^{\downarrow\uparrow} & \varepsilon_1^\downarrow + S_{11}^{\downarrow\downarrow} & T_{12}^{\downarrow\downarrow} + S_{12}^{\downarrow\downarrow} & \dots & T_{1n}^{\downarrow\downarrow} + S_{1n}^{\downarrow\downarrow} \\ S_{12}^{\downarrow\uparrow} & S_{22}^{\downarrow\uparrow} & \dots & S_{2n}^{\downarrow\uparrow} & T_{21}^{\downarrow\downarrow} + S_{21}^{\downarrow\downarrow} & \varepsilon_2^\downarrow + S_{22}^{\downarrow\downarrow} & \dots & T_{2n}^{\downarrow\downarrow} + S_{2n}^{\downarrow\downarrow} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{n1}^{\downarrow\uparrow} & S_{n2}^{\downarrow\uparrow} & \dots & S_{nn}^{\downarrow\uparrow} & T_{n1}^{\downarrow\downarrow} + S_{n1}^{\downarrow\downarrow} & T_{n2}^{\downarrow\downarrow} + S_{n2}^{\downarrow\downarrow} & \dots & \varepsilon_n^\downarrow + S_{nn}^{\downarrow\downarrow} \end{array} \right) \begin{pmatrix} c_1^\uparrow \\ c_2^\uparrow \\ \vdots \\ c_n^\uparrow \\ c_1^\downarrow \\ c_2^\downarrow \\ \vdots \\ c_n^\downarrow \end{pmatrix}$$

$$(23)$$

where

$$S_{ij}^{\sigma_1\sigma_2} = \langle \psi_i^{\sigma_1} | \hat{\mathcal{H}}^{soc} | \psi_j^{\sigma_2} \rangle \qquad T_{ij}^{\sigma\sigma} = -i\hbar \langle \psi_i^\sigma | \frac{\mathrm{d}}{\mathrm{d}t} | \psi_j^\sigma \rangle \qquad (24)$$

Expand the wavefunctions in the "*spin-adiabatic* representation"

$$|\Phi\rangle = \sum_n c_n |\psi_n\rangle = \sum_n c_n \begin{pmatrix} |\psi_n^\uparrow\rangle \\ |\psi_n^\downarrow\rangle \end{pmatrix}; \qquad (\hat{\mathcal{H}}^0 + \hat{\mathcal{H}}^{soc})|\psi_n\rangle = \varepsilon_n |\psi_n\rangle \qquad (25)$$

and substitute eq. (25) into TDKS, we have

$$\frac{\partial}{\partial t} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = -i\hbar^{-1} \begin{pmatrix} \varepsilon_1 & T_{12} & \dots & T_{1n} \\ T_{21} & \varepsilon_2 & \dots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{n1} & T_{n2} & \dots & \varepsilon_n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \qquad (26)$$

where

$$T_{ij} = -i\hbar \left(\langle\psi_i^\uparrow| \ \langle\psi_i^\downarrow| \right) \frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} |\psi_j^\uparrow\rangle \\ |\psi_j^\downarrow\rangle \end{pmatrix} \qquad (27)$$

- No spin up/down in this case.
- No SOC matrix elements needed, already included in ε_n.

- Straightforward with existing NAMD code.

Hefei-NAMD Installation

- Installation is simple!

```
git clone https://github.com/QijingZheng/Hefei-NAMD
cd Hefei-NAMD/src/namd
make clean && make
cp namd ~/bin        # make sure $HOME/bin is in you PATH
```
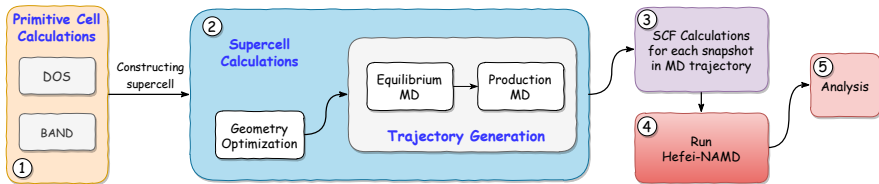
- An example compilaton screenshot:

Figure – Basic procedure of Hefei-NAMD.

- From step 1 to 2, a *LARGE* enough supercell should be chosen.

- Step 1 – 3 are done by *DFT codes*, e.g. `VASP`.

- In Step 2, we use MD to generate initial conditions for trajectory production run. Other possible approach is to sample the *Wigner distribution*,[10] which involves normal mode calculation.

- From step 2 – 3, the BZ is *only sampled at Γ-point,* which will *speedup* the calculation and *reduce* the size of wavefunction files.

- Step 4 reads results from step 3 and generate NACs for NAMD calculation.
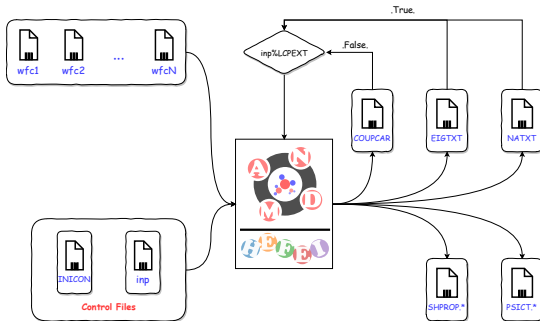
[10] *Int. J. Quantum Chem.*, **116**, 762 (2016)

Figure – Files read and written by Hefei-NAMD.

- Wavefunctions files `wfcN`, e.g. WAVECARs in `VASP`, are read from DFT calculations.

- The file `inp` and `INICON` are control files:

  - `inp` contains input parameters, e.g. basis range, temperature etc.

  - `INICON` includes initial conditions, e.g. initial band index from which charge start to relax.

- `COUPCAR` and `NATXT/EIGTXT` all contain NACs.

  - `COUPCAR` is a *binary* file and contains ALL NACs.

  - `NATXT/EIGTXT` are *text* files, as the names suggest, and ONLY contain NACs of selected basis.

  - When `LCPEXT` in `inp` is `.TRUE.`, the program will read NACs from `*TXT` instead of `COUPCAR`.

- `SHPROP.*` contain time-dependent populations of each basis state.

```
Tutorial
├── 1_GS_CALC
│       ├── 1_OPT
│       ├── 2_SCF
│       ├── 3_BND
│       └── 4_MKSC
├── 2_MD_CALC
│       ├── 1_OPT_SC
│       ├── 2_NVT_MD
│       └── 3_NVE_MD
├── 3_WAVEPROD_CALC
│       ├── 1_INIT
│       └── 2_RUN
└── 4_NAMD_CALC
        └── NAMD
```

- The example in this tutorial was carried out by *Mr. Yunzhe Tian* (yunzhe@mail.ustc.edu.cn) and published on *J. Phys. Chem. Lett.*, **11**, 586–590 (2020). A webpage version of this tutorial can also be found at

  http://staff.ustc.edu.cn/~zqj/posts/
  Hefei-NAMD-Training/

Mr. Yunzhe Tian
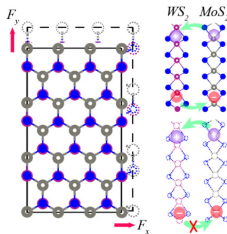
- The directory tree of the tutorial is shown left, where the number before each directory name is in strict order.

- As step $1 - 3$ are performed by DFT codes and will vary between different codes, therefore we will not cover too much about this part.

It is usually a good practice to inspect the band structure before NAMD calculations!
Check `Tutorial/1_GS_CALC/3_BND` for this part.

```
pyband  \
    # the atom index of WS2
    --occ '1 2 6' \
    # use the color stripes to plot the band
    --occL \
    # choose the colormap "seismic"
    --occLC_cmap seismic \
    # set the color range from 0 to 1
    --occLC_cbar_ticks '0 1' \
    --occLC_cbar_vmin 0 \
    --occLC_cbar_vmax 1 \
    # 0 corresponds to MoS2, 1 corresponds to WS2
    --occLC_cbar_ticklabels 'MoS$_2$ WS$_2$'
    # the name of the high-symmetry k-point
    -k mgkm
```
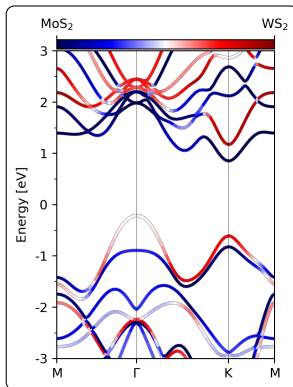

- Obviously type-II band alignment!

- How about the charge transfer process from CBM_WS2@K to CBM_MoS2@K?

https://github.com/QijingZheng/pyband

Choose supercell so that desired *k*-points of *primitive BZ* are folded on Γ-point of *supercell BZ*!

- Constructing supercell [11]

```python
#!/usr/bin/env python
from ase.io import read, write
from ase.build.supercells import make_supercell

# The transformation matrix between SC A and PC B:
# A = M B^T   ---> M = A inv(B^T)
M  = [[3, 0, 0], [3, 6, 0], [0, 0, 1]]
# primitive cell geometry stored in "primitive.vasp"
pc = read('primitive.vasp')
sc = make_supercell(pc, M)

write('POSCAR', sc, vasp5=True, sort=True, direct=True)
```

- Check if the desired *k*-point is folded on Γ-point [12]

```python
#!/usr/bin/env python
import numpy as np
from unfold import find_K_from_k

M    = np.array([[3, 0, 0], [3, 6, 0], [0, 0, 1]])
k    = [1./3, 1./3, 0]
K, _ = find_K_from_k(M, k)

if np.allclose(K, [0,0,0]):
    print("Chosen k-point folds on Gamma-point!")
```

---

[11] http://staff.ustc.edu.cn/~zqj/posts/Transforming-between-supercell-and-primitive-cell/
[12] https://github.com/QijingZheng/VaspBandUnfolding

1. Geometry optimization: to find an energy minimum configuration.
   - Supercell should be large enough because BZ is only sampled at Γ point.
2. Molecular dynamics equilibration run — canonical ensemble (NVT),

Specified Temperature
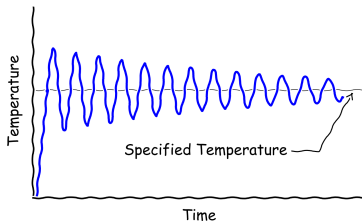
usually by velocity rescaling method, e.g. set `SMASS = −1` in VASP.
   - considered equilibrated when temperature fluctuation [13]

$$\frac{\langle \Delta T^2 \rangle}{T^2} = \frac{\langle (T_k - T)^2 \rangle}{T^2} = \frac{2}{3N}$$

   - check the configuration before proceeding to next step.
3. Molecular dynamics production run — microcanonical ensemble (NVE).
   - The equilibrated configuration i.e. `CONTCAR` of the last step is used as input of this run.
   - The duration of MD is determined by the specific problem.

---

[13] http://staff.ustc.edu.cn/~zqj/posts/NVT-MD/

Example INCAR for equilibration MD

```
General
   SYSTEM   = Your Job Name
   PREC     = Med
   ISPIN    = 1
   ISTART   = 0
   ICHARG   = 1

Electronic relaxation
   NPAR     = 4
   ISMEAR   = 0
   SIGMA    = 0.1
   ALGO     = Fast
   NELMIN   = 4
   NELM     = 120
   EDIFF    = 1E-6

Molecular Dynamics
   ISYM     = 0       # turn off symmetry for MD
   IBRION   = 0       # turn on MD
   NSW      = 500     # No. of ionic steps
   POTIM    = 1       # time step 1.0 fs
   SMASS    = -1      # velocity rescaling
   NBLOCK   = 4       # velocity rescaled
                      # every NBLOCK step
   TEBEG    = 300     # start temperature
   TEEND    = 300     # end temperature

Writing Flag
   NWRITE   = 1       # make OUTCAR small
   LWAVE    = .FALSE.
   LCHARG   = .FALSE.
```

Example INCAR for production MD

```
General
   SYSTEM   = Your Job Name
   PREC     = Med
   ISPIN    = 1
   ISTART   = 0
   ICHARG   = 1

Electronic relaxation
   NPAR     = 4
   ISMEAR   = 0
   SIGMA    = 0.1
   ALGO     = Fast
   NELMIN   = 4
   NELM     = 120
   EDIFF    = 1E-6

Molecular Dynamics
   ISYM     = 0       # turn off symmetry in MD
   IBRION   = 0       # MD flag
   NSW      = 5000    # NSW*POTIM fs
   POTIM    = 1       # time step 1.0 fs
   SMASS    = -3      # Microcanonical
   NBLOCK   = 1       # XDATCAR contains
                      # positions of each step

Writing Flag
   NWRITE   = 1       # make OUTCAR small
   LWAVE    = .FALSE. # WAVECAR not needed
   LCHARG   = .FALSE. # CHG not needed
```

1. Extract snapshots from previous MD at step 2.3 and perform SCF calculation for each one. The script `init.sh` can be used for this purpose. Be sure to prepare the input files for VASP in advance.

```bash
#!/bin/bash

# "../../2_MD_CALC/3_NVE_MD/XDATCAR" file contains trajectory for last MD run
sed -i 's/^\s*$/Direct configuration=/' ../../2_MD_CALC/3_NVE_MD/XDATCAR
python extract_positions_from_nve.py

if [ $? -eq 0 ]; then

    inp_dir=$(pwd)

    for xx in ../2_RUN/*/
    do
        cd ${xx}
        # There three files must be prepared in advance
        ln -sf ../../1_INIT/INCAR
        ln -sf ../../1_INIT/KPOINTS
        ln -sf ../../1_INIT/POTCAR
        cd ${inp_dir}
    done

fi
```

2. Choose how many snapshots you want to extract by setting NSCF (default to 2000) in the python script `extract_positions_from_nve.py`.

3. Once done, you will get NSCF directories from 0001 to 2000 under 3_WAVEPROD_CALC/2_RUN. Perform SCF calculation in each one.

- After all the SCF calculations are finished, time-dependent Kohn-Sham can be plotted with the help of the script: 3_WAVEPROD_CALC/tdksen.py

```
nsw       = 2000      # NSCF in step 3.1
dt        = 1.0       # time step in step 2.3
nproc     = 8
prefix    = '2_RUN/'
runDirs = [
    prefix + '/{:04d}'.format(ii + 1)
    for ii in range(nsw)
]
whichS    = 0         # spin index
whichK    = 0         # k-point index
# the index of WS2 atoms, starting from 0
whichA    = np.arange(54) + 54
Alabel    = r'WS$_2$'
Blabel    = r'MoS$_2$'
```
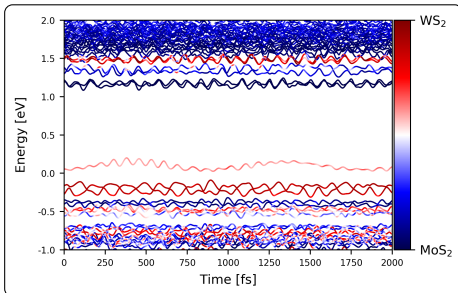

- The whole system is divided in two parts: A and B, where `whichA` contains the index of atoms within A part, the remaining atoms are default to B.

❶ Prepare input files for NAMD calculations, for example,

inp

```
&NAMDPARA
  BMIN       = 325      ! bottom band index
  BMAX       = 340      ! top band index
  NBANDS     = 388      ! number of bands

  NSW        = 2000     ! NSCF in step 3.1
  POTIM      = 1        ! MD time step
  TEMP       = 100      ! temperature

  NSAMPLE    = 100      ! number of samples
  NAMDTIME   = 1000     ! time for NAMD run
  NELM       = 1000
! electron time step
  NTRAJ      = 5000     ! SH trajectories
  LHOLE      = .FALSE. ! hole/electron SH

  RUNDIR     = "../../3_WAVEPROD_CALC/2_RUN"
  LCPEXT     = .TRUE.
/ ! NOTE THE SLASH HERE!
```
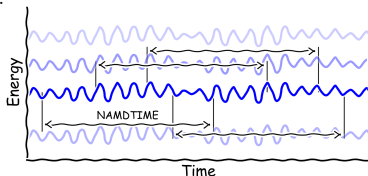
INICON

```
  87   329
 519   330
  13   330
 533   329
 541   329
 542   329
 558   329
  59   329
  62   329
  65   329
...
```

- Column 1: initial time.
- Column 2: initial band index.
- No. of rows: NSAMPLE in inp.

❷ The program will first calculate non-adiabatic couplings (NACs) and store the NACs in the file COUPCAR. If there is already the COUPCAR, then the program will read COUPCAR and perform NAMD calculatons.

For tutorial purpose, we set LCPEXT = .TRUE. and read the two *TXT files instead. *Remember* to create an empty COUPCAR!

Once we have all the needed files, type `namd` at the command line (Remember we put the executable under $HOME/bin in the installation step).

Figure – Screenshot of a Hefei-NAMD run.

When `namd` is finished, use the script `4_NAMD_CALC/NAMD/poen_fssh.py` to plot the time-dependent populations and average charge energy.
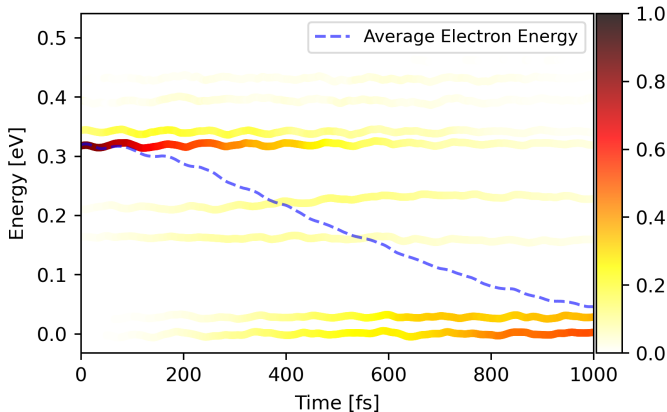

Figure – Time-dependent populations of each states shown as color stripes and average charge energy shown as dashed line.

- As is known, `WAVECAR` in `VASP` only contains *pseudo*-wavefunctions. To consider *all-electron* wavefunctions, CA-NAC developed by Dr. Weibin Chu (`wbchu@fudan.edu.cn`) come to rescue.

- For excition dynamics, refer to Dr. Xiang Jiang (`jxiang@mail.ustc.edu.cn`).

- For spin dynamics, contact Mr. Zhenfa Zheng (zzfgjs@mail.ustc.edu.cn).

Thank you!