Institute of Physics, CAS

中国科学院大学
University of Chinese Academy of Sciences

$O(N)$ method in OpenMX and its application in twisted bilayer systems

Yujin Jia
Supervisor: Hongming Weng, Quansheng Wu
Institute of Physics, Chinese Academy of Sciences

2023.11.09

Content

1 Twisted Moire Cell Problem

2 Two-step Scheme with OpenMX

3 Application: Twisted Bilayer System

4 Summary and Outlook

Twisted Moire Cell

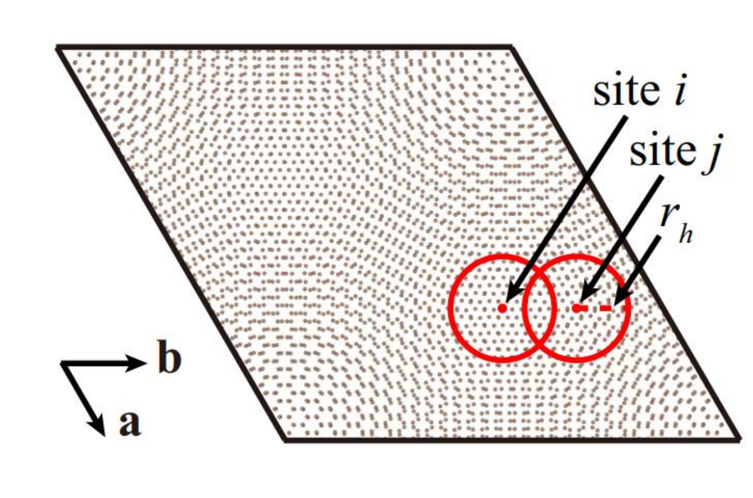
Coincidence Lattice Condition:

$$\mathbf{v}_1 = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 = m_1\mathbf{b}_1^r + m_2\mathbf{b}_2^r = \mathbf{v}_2^r$$
$$\mathbf{b}_i^r = \mathbf{b_i}\mathbf{R}_\theta^{\mathrm{T}}$$

Local translation symmetry is broken, while long range translation symmetry emerges.

Too large supercell, too many atoms, eg:
3.89° twisted $MoTe_2$ moire cell: 1302 atoms
AB stack $MoTe_2/WSe_2$ moire cell: 1095 atoms

Bring too much computational cost……



OpenMX-$O(N)$ method:

Truncate the hopping, divide the problem into small clusters to decrease computational cost.

Content

1 Twisted Moire Cell Problem

2 Two-step Scheme with OpenMX

3 Application: Twisted Bilayer System

4 Summary and Outlook

Why OpenMX

1、 Use OpenMX to perform SCF calculation.

OpenMX is based on localized pseudoatomic orbitals (PAOs) and a pseudopotential method. This scheme has natural advantages for giving tight-binding Halmiltonian.

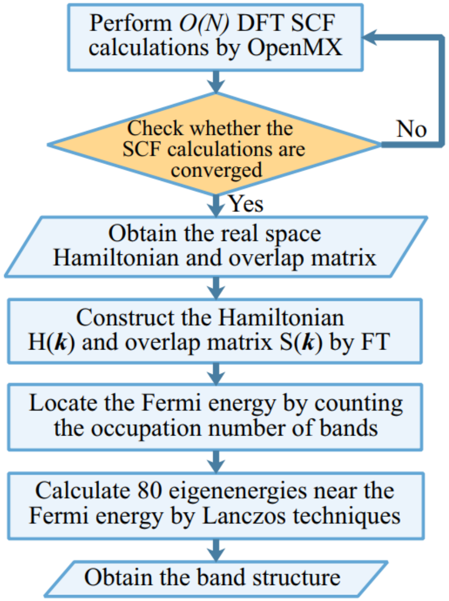
2、Use $O(N)$ Krylov subspace method in OpenMX.

Truncate the sphere near the center atom, then calculate it in chunks with much smaller computational cost.

3、Getting H(k) relatively easier than VASP.

Read from *.scfout file, just need a Fourier transform.

4、 Use Lanczos + shift inverse, get target bands.



Perform $O(N)$ DFT SCF calculations by OpenMX

Check whether the SCF calculations are converged

No

Yes

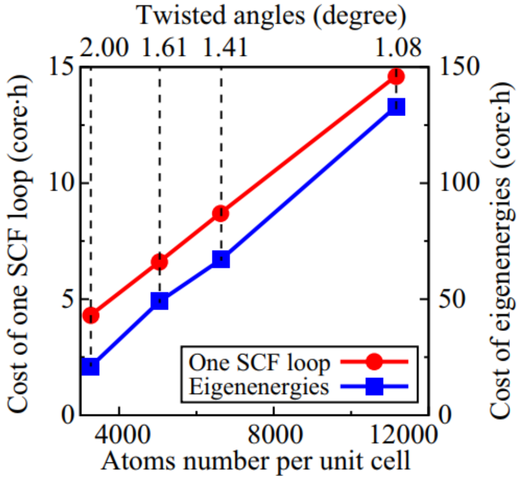
Obtain the real space Hamiltonian and overlap matrix

Construct the Hamiltonian H($\boldsymbol{k}$) and overlap matrix S($\boldsymbol{k}$) by FT

Locate the Fermi energy by counting the occupation number of bands

Calculate 80 eigenenergies near the Fermi energy by Lanczos techniques

Obtain the band structure



Ref: PhysRevB.105.125127

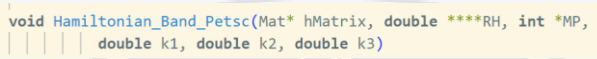
Code implementation

Read scfout:

Read Hks and iHks when SOC, read OLP

read OLPpox OLPpoy OLPpoz and density matrix, then free them

Construct OLP with PETSc

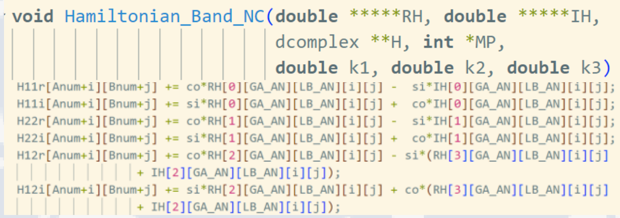


```
void Hamiltonian_Band_Petsc(Mat* hMatrix, double ****RH, int *MP,
                            double k1, double k2, double k3)
```

Find the MaxAmplitude, set sparse matrix cutoff=MaxAmplitude/10^8

Perform F.T, insert value into PETSc matrix.

Free OLP, then assemb OLP_petsc.



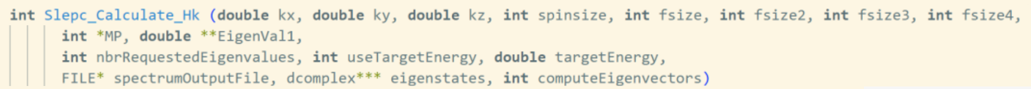
```
void Hamiltonian_Band_NC(double *****RH, double *****IH,
                         dcomplex **H, int *MP,
                         double k1, double k2, double k3)
H11r[Anum+i][Bnum+j] += co*RH[0][GA_AN][LB_AN][i][j] -  si*IH[0][GA_AN][LB_AN][i][j];
H11i[Anum+i][Bnum+j] += si*RH[0][GA_AN][LB_AN][i][j] +  co*IH[0][GA_AN][LB_AN][i][j];
H22r[Anum+i][Bnum+j] += co*RH[1][GA_AN][LB_AN][i][j] -  si*IH[1][GA_AN][LB_AN][i][j];
H22i[Anum+i][Bnum+j] += si*RH[1][GA_AN][LB_AN][i][j] +  co*IH[1][GA_AN][LB_AN][i][j];
H12r[Anum+i][Bnum+j] += co*RH[2][GA_AN][LB_AN][i][j] - si*(RH[3][GA_AN][LB_AN][i][j]
                        + IH[2][GA_AN][LB_AN][i][j]);
H12i[Anum+i][Bnum+j] += si*RH[2][GA_AN][LB_AN][i][j] + co*(RH[3][GA_AN][LB_AN][i][j]
                        + IH[2][GA_AN][LB_AN][i][j]);
```

Construct H with PETSc

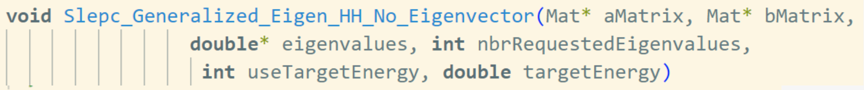
Insert value by chunks and free H in time.

Diag Hamiltonian with target energy (Ef)

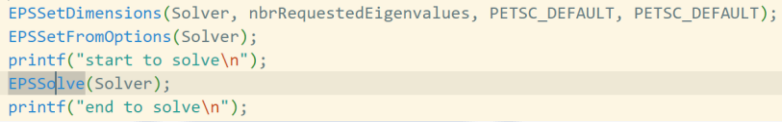
Import Slepc to calculate the eigen value near Ef



```
int Slepc_Calculate_Hk (double kx, double ky, double kz, int spinsize, int fsize, int fsize2, int fsize3, int fsize4,
     int *MP, double **EigenVal1,
     int nbrRequestedEigenvalues, int useTargetEnergy, double targetEnergy,
     FILE* spectrumOutputFile, dcomplex*** eigenstates, int computeEigenvectors)
```



```
void Slepc_Generalized_Eigen_HH_No_Eigenvector(Mat* aMatrix, Mat* bMatrix,
                                               double* eigenvalues, int nbrRequestedEigenvalues,
                                                int useTargetEnergy, double targetEnergy)
```



```
EPSSetDimensions(Solver, nbrRequestedEigenvalues, PETSC_DEFAULT, PETSC_DEFAULT);
EPSSetFromOptions(Solver);
printf("start to solve\n");
EPSSolve(Solver);
printf("end to solve\n");
```

Sort the eigen value and output

Content

1 Twisted Moire Cell Problem

2 Two-step Scheme with OpenMX

3 Application: Twisted Bilayer System

4 Summary and Outlook

Twisted $MoTe_2$: crystal structure

Bulk: $a = 3.519\ \ c = 13.976$ **Monolayer:** $a = 3.518\ \ c = 13.976$

Bilayer: $a = 3.5228\ \ c = 13.976$ (in unit of Angstrom)

AB stacking: the stable one

AA stacking: with large interlayer distance

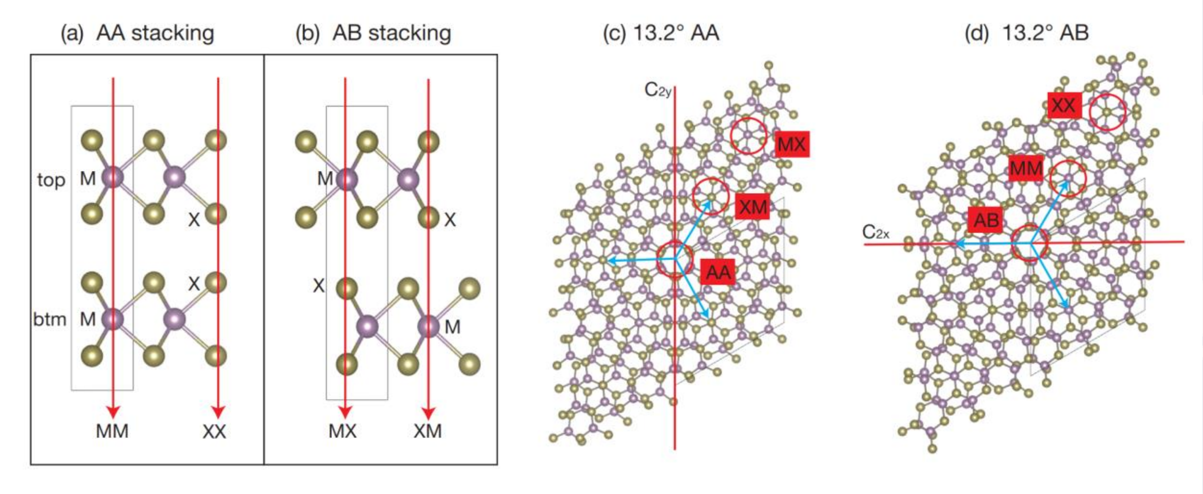
Moire structure:

Various twist angles from coincidence lattice method:

$13.2°, 9.43°, 7.34°, 4.41°, 3.89°, 3.48°$

AA configuration: C_{3z}, C_{2y}

AB configuration: C_{3z}, C_{2x}



Content

1 Twisted Moire Cell Problem

2 Two-step Scheme with OpenMX

3 Application: Twisted Bilayer System

4 Summary and Outlook

Summary and Outlook

Twisted bilayer and multilayer system with novel electronic states bring too much computational cost to DFT calculation.

OpenMX with $O(N)$ method deals with this problem efficiently.

Lanczos + shift inverse can reduce computational cost further based on OpenMX SCF Hamiltonian and Overlap.

Outlook:

1、Continue to optimizing the code

decrease the memory cost;
optimize the parallel computing…

2、Find new way to locate the E-fermi accurately

中国科学院物理研究所

Institute of Physics Chinese Academy of Sciences