Supporting information

Tunable Schottky Contacts in Hybrid Graphene/Phosphorene Nanocomposite

Wei Hu,[1,2,*] TianWang,[3] and Jinlong Yang[2,4,*]

[1]*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

[2]*Hefei National Laboratory for Physical Sciences at Microscale, University of Science and Technology of China, Hefei, Anhui 230026, China*

[3]*Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei, Anhui 230026, China*

[4]*Synergetic Innovation Center of Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei, Anhui 230026, China*

Email: whu@lbl.gov and jlyang@ustc.edu.cn

In order to simulate hybrid composite structures of 2D van der Waals heterojunctions made layer by layer, we need to choose two special supercells for each 2D material with different crystal systems and lattice constants, making them have a smaller lattice mismatch for each other. We have written a small program named LatticeMatch to construct theoretical models for 2D van der Waals heterojunctions with a smaller lattice mismatch for different 2D materials. Its basic idea is based on the rotation matrix of extended orthogonal lattices for different 2D materials to search their minimum matching supercell models in 2D van der Waals heterojunctions.

! Author: Wei Hu
! E-mail: gyrw4588@mail.ustc.edu.cn
! Google Scholar Citations: http://scholar.google.com/citations?user=t8Ncyb4AAAAJ

! The LatticeMatch code is used to achieve lattice matching for different 2D systems.
! Its basic idea is based on the rotation matrix of orthogonal lattice ot search the
! minimum matching model.

```
program LatticeMatch

   implicit none
   integer i,j,ii,jj,k,kk,kki,M,N,Nmax
   parameter    (Nmax=15)

   real PI

   parameter    (PI=3.14159265358979323846)


   real Gx(Nmax),Gy(Nmax),Cx(Nmax),Cy(Nmax),Test(Nmax)
   real Cax(Nmax,Nmax),Cby(Nmax,Nmax),Rx(Nmax),Ry(Nmax)

   real Ga,Gb,Ca,Cb,L,Lx,Ly,Lg,Lc,a,Aa,Aaa,b

   Ga=2.4690
   Gb=4.2760 ! Orthogonal lattice for graphene
   Ca=3.2981
   Cb=4.6232 ! Orthogonal lattice for phosphorene
```

```
a=0.01 ! The minimum angle of rotation
b=0.25 ! Error control parameter

M=180/a

do i=1, Nmax
   Gx(i)=0.00
   Gy(i)=0.00
   Cx(i)=0.00
   Cy(i)=0.00
   Rx(i)=0.00
   Ry(i)=0.00
   Test(i)=0.00
enddo

do i=1, Nmax
   do j=1, Nmax
      Cax(i,j)=0.00
      Cby(i,j)=0.00
   enddo
enddo

N=(Nmax+1)/2


if (N.gt.1)then

   do i=1, N-1
      Gx(i)=0.00-Ga*(N-i)
      Gy(i)=0.00-Gb*(N-i)
      Cx(i)=0.00-Ca*(N-i)
      Cy(i)=0.00-Cb*(N-i)
   enddo

endif

do i=N, Nmax
   Gx(i)=Ga*(i-N)
   Gy(i)=Gb*(i-N)
   Cx(i)=Ca*(i-N)
   Cy(i)=Cb*(i-N)
enddo
```

```
 do i=1, Nmax
    do j=1, Nmax

       write(2,"(1X,a3,i2,a2,f8.4,a4,i2,a2,f8.4)")
&                        "Gx(",i,")=",Gx(i)," Gy(",j,")=",Gy(j)

       write(3,"(1X,a3,i2,a2,f8.4,a4,i2,a2,f8.4)")
&                        "Cx(",i,")=",Cx(i)," Cy(",j,")=",Cy(j)

    enddo
 enddo


 do k=1,M !M=180/a

    Aa=(k-1)*a
    Aaa=Aa*(2*PI/360)

    do i=1, Nmax
       do j=1, Nmax
          Cax(i,j)=0.00
          Cby(i,j)=0.00
       enddo
    enddo

 ! Rotating the lattice

    do i=1, Nmax
       do j=1, Nmax

          Lc=sqrt(Cx(i)*Cx(i)+Cy(j)*Cy(j))


          if(Lc.gt.0.0001)then

             Cax(i,j)=Lc*(cos(Aaa)*(Cx(i)/Lc)-sin(Aaa)*(Cy(j)/Lc))
             Cby(i,j)=Lc*(sin(Aaa)*(Cx(i)/Lc)+cos(Aaa)*(Cy(j)/Lc))

          endif

       enddo
    enddo

    write(4,"(1X,a3,f8.4)")"Aa=",Aa
```

```
      do i=1, Nmax
        do j=1, Nmax

          write(4,"(1X,a4,i2,a1,i2,a2,f8.4,a5,i2,a1,i2,a2,f8.4)")
     &          "Cax(",i,",",j,")=",Cax(i,j)," Cby(",i,",",j,")=",Cby(i,j)

        enddo
      enddo

      kk=0

      do i=1, Nmax
        do j=1, Nmax


          do ii=1, Nmax
            do jj=1, Nmax


              Lx=Gx(i)-Cax(ii,jj)
              Ly=Gy(j)-Cby(ii,jj)

              L=sqrt(Lx*Lx+Ly*Ly)

              if(L.lt.b)then

                kk=kk+1

                Rx(i)=Gx(i)
                Ry(j)=Gy(j)

                Test(kk)=0.00

            if((abs(Gx(i)).gt.0.0001).and.(abs(Gy(j)).gt.0.0001)) then
              Test(kk)=Gx(i)/Gy(j)
            endif

                write(7,"(1X,a3,i4)")"kk=",kk
                write(7,"(1X,a5,i4,a2,f6.2)")"Test(",kk,")=",Test(kk)
                write(7,"(1X,a3,f8.4)")"Aa=",Aa
                write(7,"(1X,a3,i2,a2,f8.4,a4,i2,a2,f8.4)")
     &                "Gx(",i,")=",Rx(i)," Gy(",j,")=",Ry(j)
```

```
                endif

              enddo
            enddo

        enddo
    enddo

! Output qualified lattice matching angle

    if(kk.ge.7)then

        write(*,"(1X,a3)")"OK!"
        write(*,"(1X,a3,f8.4)")"Aa=",Aa
        write(*,"(1X,a3,i4)")"kk=",kk

        do kki=1, kk
           write(*,"(1X,a5,i4,a2,f6.2)")"Test(",kki,")=",Test(kki)
        enddo

        write(8,"(1X,a3)")"OK!"
        write(8,"(1X,a3,f8.4)")"Aa=",Aa
        write(8,"(1X,a3,i4)")"kk=",kk

        do kki=1, kk
           write(8,"(1X,a5,i4,a2,f6.2)")"Test(",kki,")=",Test(kki)
        enddo


    endif

enddo ! do k=1,M

end
```