```
In[279]:= Print["  Figure3cd(Layers1L15nmRIn1d14.m, Figure3cd(Layers2L15nmRIn1d14.m"];
Print[];
Print["   The source of the data of the manuscript"];
Print["   'Structure and energetics of carbon, "];
Print["   hexagonal boron nitride, and"];
Print["   carbon/hexagonal boron nitride"];
Print["   single-layer and bilayer nanoscrolls' "];
Print[" / A.I. Siahlo, N.A. Poklonski, A.V. Lebedev,"];
Print[" I.V. Lebedeva, A.M. Popov, S.A. Vyrko, "];
Print[" A.A. Knizhnik, Yu.E. Lozovik "];
Print[" // Phys. Rev. Materials.— 2018.— V. 2,"];
Print[" № 3.— P. 036001 (9 pp.)."];
Print[" [DOI: 10.1103/PhysRevMaterials.2.036001] "];
Print[" ----------------------------------------------------"];

NoL1 = 1; NoL2 = 2;
NoLp = NoL1;
Print[" I.0 The Units (nm, meV, AA)"];
"nm=10^(-9)m;";
nm = 10^(-9) m;
AA = 10^(-10) m;
JJkgms = kg m^2/s^2;
Cl = Amper s;
"eV=JJ Electronp;";
JJ = eV/Electron;
JJms = (kg m^2)/s^2;
meV = N[eV/1000];
Print["----------------------------------------------------------"];

Print[" I.1. All Input Parameters and Constants--------------"];
Print[" I.1.1. The sampling parameters"]
npRIn1 = 1000;
Print["npRIn1=", npRIn1];
Print[" I.1.2. The Input Geometry Parameters of the system"];
L14d839nm = 14.839 nm;
L12d709nm = 12.709 nm;
L129d678nm = 29.678 nm;
L1p = L129d678nm;
L1p = L14d839nm;

Print[" The carbon nanoribbon length L1=", L1p/nm, "nm"];
Lw11d8nm = 11.8 nm;
Lwp = Lw11d8nm;
Print[" The carbon nanoribbon width Lw=", Lwp/nm, "nm"];
Print["----------------------------------------------------------"];

Print[" Number of the layers in carbon nanoscroll NoL=", NoLp];
Print[" The length of a carbon nanoribbon L1=", L1p/nm, "nm"];
Lw1nm = 1. nm; Lwp = Lw1nm;
Print[" The carbon nanoribbon width Lw=", Lwp/nm, "nm"];
RIn1d1nm = 1.1 nm;
RIn1d2nm = 1.2 nm;
RIn1d14nm = 1.14 nm;
RIn2nm = 2.047 nm;
RIn2d1nm = 2.1 nm;
RIn2d2nm = 2.2 nm;
RIn2d3nm = 2.3 nm;
```

```
RIn2d4nm = 2.4 nm;
RIn2d5nm = 2.5 nm;
RIn2d6nm = 2.6 nm;
RIn1p = RIn2d5nm;
RIn1p = RIn2d3nm;
RIn1p = RIn2d2nm;
RIn1p = RIn2d1nm;
RIn1p = RIn1d14nm;
Print[" The inner radius of the nanoscroll RIn1=", RIn1p/nm, "nm"];
Print[" I.1.2. The Input Energy Constants"];
Print[" epsVdW - the interlayer interaction energy per one atom of"];
Print[" the nanoscroll:"];
epsVdW35 = 35.0 meV/atom; epsVdWp = epsVdW35;
Print[" epsVdW=", epsVdWp/(eV/atom), "eV/atom"];
Print[" C - the bending elastic constant:"];
C201 = 2.01 eV AA^2/atom;
CBN1328 = 1.328 eV AA^2/atom;
CCp = C201;
CBNp = CBN1328;
CBNp = CCp;
Print[" CCelast=", CCp/(eV AA^2/atom), "eV AA^2/atom"];
Print[" CCBNelast=", CBNp/(eV AA^2/atom), "eV AA^2/atom"];
Print[" I.1.3.The Input Geometry constants--------------"];
Print[" The interatomic distance aCC and the interlayer distance h"];
aCC142AA = 1.42 AA; aCCp = aCC142AA;
h335nm = 0.3354 nm; hp = h335nm;
Print["h=", hp/nm, " nm  (Interlayer distance)"];
Print[" aCC=", aCCp/nm, "nm, h=", hp/nm, "nm"];
NatomsInCell2 = 2; NatomsInCellp = NatomsInCell2;
Print["NatomsInCell=", NatomsInCellp];

Print[" dPhi12 - The difference of the inner angles of the spirales"];
Print["   of the Layers"];
dPhi12eq0 = 0.0 Pi;
dPhi12eqPi = 1.0 Pi;
dPhi12p = 0.0 Pi;
dPhi12p = 1.0 Pi;
dPhi12p = 0.5 Pi;
Print[" I.4.The parameters for the visualisation"];
RIn1MinMonoScroll = hp/5;
RIn1MinBiScroll = hp/5;
RIn1MaxMonoScroll = 4 nm;
RIn1MaxBiScroll = 8 nm;
PlotRangeMonoScroll = {-4 eV/atom, 12 eV/atom};
PlotRangeBiScroll = {-10 eV/atom, 30 eV/atom};
ShowSpirales = True;
ShowThePlot = True;

Print[" I.5. The parameters of visualization that depend on NoL=",NoLp];
PlotRangep = Switch[NoLp, 1, PlotRangeMonoScroll, 2, PlotRangeBiScroll];
RIn1Minp = Switch[NoLp, 1, RIn1MinMonoScroll, 2, RIn1MinBiScroll];
RIn1Maxp = Switch[NoLp, 1, RIn1MaxMonoScroll, 2, RIn1MaxBiScroll];
PlotRangep = Switch[NoLp, 1, PlotRangeMonoScroll, 2, PlotRangeBiScroll];
RIn1Maxp = Switch[NoLp, 1, RIn1MaxMonoScroll, 2, RIn1MaxBiScroll];
tL1 = Switch[NoLp, 1, {7. nm, 10. nm, 12.5 nm, 15. nm},
      2, {15. nm, 20. nm, 25. nm, 30. nm}];
```

```
Print[" I.6. The parameters of the output file"];
NanoscrollNamep =
  StringJoin["Nanoscroll", ToString[NoLp], "L", ToString[L1p/nm],
   "nm"];
Print[" NanoscrollName=", NanoscrollNamep];
CarbonNanoscrollEnergyVsRInFileName =
  StringJoin[NanoscrollNamep, ".txt"];
Print[CarbonNanoscrollEnergyVsRInFileName];
Print[" (The output of the data to a file Is Not Performed)"];
npRIn1 = 1000;
Print[" The number of the output points = ", npRIn1];
Print[" I.7. The Input Numerical Constants used in the programm"];
Print[" The Indexes used for the work with EVdW[...] function"];
iEVdW = 1; iEVdW1Un1 = 2; iEVdW1Ov1 = 3; iEVdW1Un2 = 4; iEVdW1Ov2 = 5;
iEVdW2Un1 = 6; iEVdW2Ov1 = 7;
Print[" --------End of the Input---------------"];
AA = 0.1 nm; PhiIn := φIn; PhiOut := φOut;
Print["----------------------------------------------------------"];

Print[" II. The derivated parameters and the functions required"];
Print[" II.1. The derivated parameters"];
fSa[aCC_] := aCC^2 3 Sqrt[3]/4; fSa[aCCp]; Sap = fSa[aCCp];
Print["  The cell area Sa=", fSa[aCC], "=", Sap/nm^2, "nm^2"];
Print[" II.2. The required functions--------------"];
Print[" II.2.1. The function fSpiraleLen[", NoLp,",φIn, φOut, h] defines"];
Print["  the Length of a Spirale with the inner agle φIn and the outer angle φOut(>=φIn):"
fSpiraleLen[NoLv_, PhiInv_, PhiOutv_, hv_] :=
   (1/(4 Pi) hv NoLv (-PhiInv Sqrt[1 + PhiInv^2] + PhiOutv Sqrt[1 + PhiOutv^2] - ArcSinh[P
Print["  fSpiraleLen[", NoLp, ", φIn, φOut, h]=",
  fSpiraleLen[NoLp, PhiIn, PhiOut, h], "."];
Print[" II.2.2. The function fElast[φIn,ρOut] is required to calculate an nanoscrollelasti
fElast[PhiInv_,
      PhiOutv_] := (Sqrt[PhiInv^2 + 1]/PhiInv -
        Sqrt[PhiOutv^2 + 1]/PhiOutv - ArcSinh[PhiInv] +
    ArcSinh[PhiOutv]);
Print[" fElast[φIn,φOut] = ",
  fElast[PhiIn, PhiOut], "."];
Print[" II.2.3. Define the function fPhiOutvsPhiInLh[", NoLp, ",", PhiIn, ",L,h]."]
fPhiOutvsPhiInLh[NoLv_, PhiInv_, Lv_, hv_] :=
    Sqrt[4 π Lv/(NoLv hv) + PhiInv^2];
Print[" The function fPhiOutvsPhiInLh[", NoLp, ",", PhiIn, ",L,h]=",
  fPhiOutvsPhiInLh[NoLp, PhiIn, L, h], " is a
   good approximation to obtain the value of φOut for the defined φIn,L,h."];
fPhiInvsPhiOutLh[NoLv_, PhiOutv_, Lv_, hv_] := Sqrt[PhiOutv^2 - 4 Pi Lv/(NoLv hv)];
Print[" The inverse function fPhiInvsPhiOutLh[", NoLp,",φOut, L, h]]=",fPhiInvsPhiOutLh[No
Print[" could be used in the program applications if ROut (instead of RIn) is the input pa

Print[" II.2.4. The functions fSpirale1Under(Over)Spirale1Length[NoLv,PhiIn1v ,PhiOut1v,hv
fSpirale1UnderSpirale1Length[NoLv_, PhiIn1v_ , PhiOut1v_, hv_] := fSpiraleLen[NoLv, PhiIn1
fSpirale1OverSpirale1Length[NoLv_, PhiIn1v_ , PhiOut1v_, hv_] := fSpiraleLen[NoLv, PhiIn1v

fSpirale1UnderSpirale2Length[NoLv_, PhiIn1v_ , PhiOut1v_, hv_, dPhi12v_] := fSpiraleLen[No
fSpirale1OverSpirale2Length[NoLv_, PhiIn1v_ , PhiOut1v_, hv_, dPhi12v_] := fSpiraleLen[NoL
fSpirale2UnderSpirale1Length[NoLv_, PhiIn1v_ , PhiOut1v_, hv_, dPhi12v_] := fSpiraleLen[No
fSpirale2OverSpirale1Length[NoLv_, PhiIn1v_ , PhiOut1v_, hv_, dPhi12v_] := fSpiraleLen[NoL

Print["  These functiona are not required, but could be helpful),"];
```

```
If[NoLp == 1,
  Print["fSpirale1UnderSpirale1Length[1,PhiIn1v ,PhiOut1v,hv]="];
  Print["  =fSpiraleLen[NoLv,PhiIn1v ,PhiOut1v-2Pi,hv]=",
   fSpiraleLen[NoLv, PhiIn1v , PhiOut1v - 2 Pi, hv], ";"];
  Print["  fSpirale1UnderSpirale1Length[NoLp,PhiIn1p,PhiOut1p,hp]=fSpirale1UnderSpirale1Le
  Print["  =",
   fSpirale1UnderSpirale1Length[NoLp, PhiIn1p , PhiOut1p, hp]/nm,
   "nm."];

  Print["fSpirale1OverSpirale1Length[1,PhiIn1v ,PhiOut1v,hv]="];
  Print["  =fSpiraleLen[NoLv,PhiIn1v+2Pi ,PhiOut1v,hv]=",
   fSpiraleLen[NoLv, PhiIn1v + 2 Pi , PhiOut1v, hv], ";"];
  Print["  fSpirale1OverSpirale1Length[NoLp,PhiIn1p ,PhiOut1p,hp]=fSpirale1OverSpirale1Len
   PhiIn1p/(2 Pi), "(2Pi),", PhiOut1p/(2 Pi), "(2Pi),", hp/nm,
   "nm] ="];
  Print["  =",
   fSpirale1OverSpirale1Length[NoLp, PhiIn1p , PhiOut1p, hp]/nm,
   "nm."];
  ];

If[NoLp == 2,
  Print["  fSpirale1UnderSpirale2Length[1,PhiIn1v ,PhiOut1v,hv,dPhi12v]="];
  Print["  fSpiraleLen[NoLv,PhiIn1v, PhiOut1v -2 Pi/NoLv,hv]=",
   fSpirale1UnderSpirale2Length[1, PhiIn1v , PhiOut1v, hv, dPhi12v],
   ";"];
  Print["fSpirale1UnderSpirale2Length[NoLp,PhiIn1p,PhiOut1p,hp,dPhi12p]=fSpirale1UnderSpir
   PhiIn1p/(2 Pi), "(2Pi),", PhiOut1p/(2 Pi), "(2Pi),", hp/nm, "nm,",
   dPhi12p/(2 Pi), "(2Pi)] ="];
  Print["  =",
   fSpirale1UnderSpirale2Length[NoLp, PhiIn1p , PhiOut1p, hp,
     dPhi12p]/nm, "nm."];
  Print["  fSpirale1OverSpirale2Length[1,PhiIn1v ,PhiOut1v,hv,dPhi12v]="];
  Print["  =fSpiraleLen[NoLv,PhiIn1v+Pi ,PhiOut1v,hv]=",
   fSpiraleLen[NoLv, PhiIn1v + Pi , PhiOut1v, hv], ";"];
  Print["fSpirale1OverSpirale2Length[NoLp,PhiIn1p ,PhiOut1p,hp]=fSpirale1OverSpirale1Lengt
   PhiIn1p/(2 Pi), "(2Pi),", PhiOut1p/(2 Pi), "(2Pi),", hp/nm,
   "nm] ="];
  Print["  =",
   fSpirale1OverSpirale2Length[NoLp, PhiIn1p , PhiOut1p, hp, dPhi12p]/
    nm, "nm."];
  Print[""];
  Print["  fSpirale2UnderSpirale1Length[1,PhiIn1v ,PhiOut1v,hv]="];
  Print["  fSpiraleLen[NoLv,PhiIn1v, PhiOut1v -2 Pi/NoLv,hv]=",
   fSpirale1UnderSpirale2Length[1, PhiIn1v , PhiOut1v, hv, dPhi12v],
   ";"];
  Print["fSpirale2UnderSpirale1Length[NoLp,PhiIn1p,PhiOut1p,hp,dPhi12p]=fSpirale2UnderSpir
   PhiIn1p/(2 Pi), "(2Pi),", PhiOut1p/(2 Pi), "(2Pi),", hp/nm, "nm,",
   dPhi12p/(2 Pi), "(2Pi)] ="];
  Print["  =",
   fSpirale1UnderSpirale2Length[NoLp, PhiIn1p , PhiOut1p, hp,
     dPhi12p]/nm, "nm."];
  Print["  fSpirale1OverSpirale2Length[1,PhiIn1v ,PhiOut1v,hv]="];
  Print["  =fSpiraleLen[NoLv,PhiIn1v+Pi ,PhiOut1v,hv]=",
   fSpiraleLen[NoLv, PhiIn1v + Pi , PhiOut1v, hv], ";"];
  Print["fSpirale1OverSpirale2Length[NoLp,PhiIn1p ,PhiOut1p,hp]=fSpirale1OverSpirale2Lengt
   PhiIn1p/(2 Pi), "(2Pi),", PhiOut1p/(2 Pi), "(2Pi),", hp/nm,
```

```
   "nm] ="];
  Print["  =",
   fSpirale1OverSpirale2Length[NoLp, PhiIn1p , PhiOut1p, hp, dPhi12p]/
    nm, "nm."];
  ];

Print[" II.2.4. The function fRIn1Sharp[NoLv,L1v,hv]"]
fRIn1Sharp[NoLv_, L1v_, hv_] := (L1v/(2 Pi) - (NoLv hv/2));
Print["fRIn1Sharp[NoLv,L1v,hv]=", fRIn1Sharp[NoLv, L1v, hv]];
Print["is a good approximation to obtain the value of the sharp in the dependence ScrollEn
Print["fRIn1Sharp[", NoLp, ", ", L1p/nm, "nm, ", hp/nm, "nm] = ",
  fRIn1Sharp[NoLp, L1p, hp]/nm, "nm"];
Print["-----------------------------------------------------------"];

Print[" III. Begin of Calculation "];
If[NoLp == 1,
  Print[" III.1. The inner and the outer angle of the spirale of the layer:"]];
If[NoLp == 2,
  Print[" III.1. The inner and the outer angles of the spirales of the layers:"]];
Print[" φIn1=", RIn1 2 Pi/(NoLp h),", φOut1=fPhiOutvsPhiInLh[", NoLp, ",ϕIn1,L1,h];"];
fPhiIn1[NoLv_, RIn1v_, hv_] := RIn1v 2 Pi/(NoLv hv);
PhiIn1p = fPhiIn1[NoLp, RIn1p, hp];
fPhiOut1[NoLv_, L1v_, RIn1v_, hv_] :=
    fPhiOutvsPhiInLh[NoLv, fPhiIn1[NoLv, RIn1v, hv], L1v, hv];
Print[" For RIn1=", RIn1p/nm, "nm,h=", hp/nm, "nm:"];
PhiOut1p = fPhiOut1[NoLp, L1p, RIn1p, hp];
ROut1p = PhiOut1p NoLp hp/(2 Pi);
Print[" φIn1=", PhiIn1p/(2 Pi), "(2Pi),ΨOut1=",
  PhiOut1p/(2 Pi), "(2Pi)."];
fPhiIn2[NoLv_, RIn1v_, hv_, dPhi12v_] :=
    fPhiIn1[NoLv, RIn1v, hv] + dPhi12v;
PhiIn2dPhi12p =
   fPhiIn2[NoLp, RIn1p, hp, 0];(*www orig 2022.10*)
PhiIn2dPhi12p =
   fPhiIn2[NoLp, RIn1p, hp,
  dPhi12p];(* for dPhi12p!=0, checked 2022.10*)
PhiIn2dPhi12Pip =
   fPhiIn2[NoLp, RIn1p, hp, Pi];
fPhiOut2[NoLv_, L1v_, RIn1v_, hv_, dPhi12v_] :=
    fPhiOutvsPhiInLh[NoLv, fPhiIn2[NoLv, RIn1v, hv, dPhi12v], L1v,
   hv];
PhiOut2dPhi12p =
    fPhiOut2[NoLp, L1p, RIn1p, hp, dPhi12p];
If[NoLp == 2,
  Print["  φIn2=", PhiIn2dPhi12p/(2 Pi),"(2Pi),φOut2=", PhiOut2dPhi12p/(2 Pi), "(2Pi)"];
  PhiOut2dPhi12Pip = fPhiOut2[NoLp, L1p, RIn1p, hp, Pi];
  Print[" for dφ12=Pi: PhiIn2=", PhiIn2dPhi12Pip/(2 Pi),
   "(2Pi),φOut2=", PhiOut2dPhi12Pip/(2 Pi), "(2Pi)"];];
Print["L1=", L1p/nm, "nm, RIn1=", RIn1p/nm, "nm"];
If[NoLp == 1, Print[" Plot the Spirale of the layer:"]];
If[NoLp == 2, Print[" Plot Spirales of the layers:"]]; "for dφ12=0";
Spirale1Plot = PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn1p, PhiOut1p},
      PlotRange -> {{-1.1 ROut1p/nm,
      1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
            1.1 ROut1p/nm}}, PlotStyle -> {Red, Thin}, Axes -> None];
If[NoLp == 1, Print[Show[Spirale1Plot]];
Print["Manipulating of Spirale1Plot for the different RIn1 and L:"];
```

```
Manipulate[PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm,
{Phiv, fPhiIn1[NoLp, RIn1nmm nm, hp],fPhiOut1[NoLp, L1nmm nm, RIn1nmm nm, hp]},
      PlotRange -> {{-1.1 ROut1p/nm,
      1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
            1.1 ROut1p/nm}}, PlotStyle -> {Red, Thin}, Axes -> None]
,{{RIn1nmm,RIn1p/nm},RIn1Minp/nm,RIn1Maxp/nm},{{L1nmm,L1p/nm},0.5 tL1[[1]]/nm,1.5 tL1[[Lenç
]
If[NoLp > 1,
 Print[" Plot the Spirale of the layers:"];
 Spirale2Plot =
  PolarPlot[(Phiv - Pi) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn2dPhi12p + Pi,
    PhiOut2dPhi12p + Pi},
       PlotRange -> {{-1.1 ROut1p/nm,
      1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
             1.1 ROut1p/nm}}, PlotStyle -> {Blue, Thin},
   Axes -> None];
 Spirale2dPhi12PiPlot =
  PolarPlot[(Phiv - Pi) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn2dPhi12p + Pi,
    PhiOut2dPhi12Pip + Pi},
       PlotRange -> {{-1.1 ROut1p/nm,
      1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
             1.1 ROut1p/nm}}, PlotStyle -> {Blue, Thin},
   Axes -> None];
 Print[Show[{Spirale1Plot, Spirale2Plot}]];
 ]
If[NoLp == 1,
    Spirale1OverSpirale1Plot =
      If[PhiIn1p + 2 Pi < PhiOut1p,
        PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn1p + 2 Pi,
            PhiOut1p},

     PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
                1.1 ROut1p/nm}}, PlotStyle -> {Red, Thick},
     Axes -> None], {}];
    Spirale1UnderSpirale1Plot =
      If[PhiIn1p < PhiOut1p - 2 Pi,
        PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn1p,
            PhiOut1p - 2 Pi},

     PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
                1.1 ROut1p/nm}}, PlotStyle -> {Red, Thick},
     Axes -> None], {}];
    Print[
   " {Spirale,Spirale1UnderSpirale1},{Spirale1,Spirale1OverSpirale1}:"];
    Print[Show[{Spirale1Plot, Spirale1UnderSpirale1Plot}],
      Show[{Spirale1Plot, Spirale1OverSpirale1Plot}]];
  ];
If[NoLp == 2,
  Spirale1UnderSpirale2dPhi120Plot =
      If[PhiIn1p < PhiOut2dPhi12p - Pi,
        PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn1p,
            PhiOut2dPhi12p - Pi}, PlotStyle -> {Red, Thick},

     PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
                1.1 ROut1p/nm}}], {}];
  Spirale1OverSpirale2dPhi120Plot =
      If[PhiIn1p + Pi < PhiOut1p,
```

```
      PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv,
          PhiIn1p + Pi +
            dPhi12p, PhiOut1p},
        PlotStyle -> {Red, Thick},

   PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
              1.1 ROut1p/nm}}], {}];
Spirale2UnderSpirale1dPhi120Plot =
    If[PhiIn2dPhi12p + Pi < PhiOut1p,
      PolarPlot[(Phiv - Pi) NoLp hp/(2 Pi)/nm, {Phiv,
          PhiIn2dPhi12p + Pi, PhiOut1p},
   PlotStyle -> {Blue, Thick},

   PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
              1.1 ROut1p/nm}}], {}];
Spirale2OverSpirale1dPhi120Plot =
    If[2 Pi + PhiIn2dPhi12p -
          dPhi12p <
        PhiOut2dPhi12p + Pi,
      PolarPlot[(Phiv - Pi) NoLp hp/(2 Pi)/nm, {Phiv,
          2 Pi + PhiIn2dPhi12p -
            dPhi12p,
          PhiOut2dPhi12p + Pi}, PlotStyle -> {Blue, Thick},

   PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
              1.1 ROut1p/nm}}], {}];
Print["Plot Spirales for dPhi12=Pi (could be NotRequired, dPhi12=0 in this program)"];
Spirale1UnderSpirale2dPhi12PiPlot =
    If[PhiIn1p < PhiOut2dPhi12Pip - Pi,
      PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv, PhiIn1p,
          PhiOut2dPhi12Pip - Pi}, PlotStyle -> {Red, Thick},

   PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
              1.1 ROut1p/nm}}], {}];
Spirale1OverSpirale2dPhi12PiPlot =
    If[PhiIn1p + Pi +
          dPhi12p < PhiOut1p,
      PolarPlot[(Phiv) NoLp hp/(2 Pi)/nm, {Phiv,
          PhiIn1p + Pi +
            dPhi12p, PhiOut1p},
        PlotStyle -> {Red, Thick},

   PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
              1.1 ROut1p/nm}}], {}];
Spirale2UnderSpirale1dPhi12PiPlot =
    If[PhiIn2dPhi12p + Pi < PhiOut1p,
      PolarPlot[(Phiv - Pi) NoLp hp/(2 Pi)/nm, {Phiv,
          PhiIn2dPhi12Pip + Pi, PhiOut1p},
   PlotStyle -> {Blue, Thick},

   PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
              1.1 ROut1p/nm}}], {}];
Spirale2OverSpirale1dPhi12PiPlot =
    If[2 Pi + PhiIn2dPhi12Pip -
          dPhi12p <
        PhiOut2dPhi12Pip + Pi,
      PolarPlot[(Phiv - Pi) NoLp hp/(2 Pi)/nm, {Phiv,
```

```
            2 Pi + PhiIn2dPhi12Pip -
              dPhi12p,
            PhiOut2dPhi12Pip + Pi}, PlotStyle -> {Blue, Thick},

     PlotRange -> {{-1.1 ROut1p/nm, 1.1 ROut1p/nm}, {-1.1 ROut1p/nm,
               1.1 ROut1p/nm}}], {}];

    Print[
   " {Spirale1,Spirale2,Spirale1UnderSpirale2,Spirale2UnderSpirale1},"];
    Print[
   "     {Spirale1,Spirale2,Spirale1OverSpirale2,Spirale2OverSpirale1}"];
    Print[" for dPhi12=0: ", Show[Spirale1Plot, Spirale2Plot],
      Show[Spirale1Plot, Spirale2Plot,
    Spirale1UnderSpirale2dPhi120Plot,
        Spirale2UnderSpirale1dPhi120Plot],
      Show[Spirale1Plot, Spirale2Plot,
    Spirale1OverSpirale2dPhi120Plot,
        Spirale2OverSpirale1dPhi120Plot]];
    Print[" for dPhi12=Pi: ",
   Show[Spirale1Plot, Spirale2dPhi12PiPlot],
      Show[Spirale1Plot, Spirale2dPhi12PiPlot,
    Spirale1UnderSpirale2dPhi12PiPlot,
        Spirale2UnderSpirale1dPhi12PiPlot],
      Show[Spirale1Plot, Spirale2dPhi12PiPlot,
    Spirale1OverSpirale2dPhi12PiPlot,
        Spirale2OverSpirale1dPhi12PiPlot]];
  ];

Print[" III.2. The nanoscroll energy calculation"];
Print[" III.2.1. The elastic energy calculation"];
fEelastCC[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_, CCv_] :=
    Module[{},
      Return[2 Pi CCv Lwv/(hv fSa[aCCv]) fElast[
              fPhiIn1[NoLv, RIn1v, hv],
              fPhiOut1[NoLv, L1v, RIn1v, hv]]];];
fEelastCBN[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_, CBNv_] :=
    Module[{},
      Return[2 Pi CBNv Lwv/(hv fSa[aCCv]) fElast[
              fPhiIn1[NoLv, RIn1v, hv],
              fPhiOut1[NoLv, L1v, RIn1v, hv]]];];
EelastCCp = fEelastCC[NoLp, Lwp, L1p, RIn1p, hp, aCCp, CCp];
EelastCBNp = fEelastCBN[NoLp, Lwp, L1p, RIn1p, hp, aCCp, CBNp];
Print[" EelastC=", EelastCCp/(eV/atom), "eV/atom"];
If[NoLp == 2, Print[" EelastBN=", EelastCBNp/(eV/atom), "eV/atom"];];
Print[" III.2.2. The Van-der-Waals energy calculation"];
"The definition of the function ";
"'fEVdWLayer1Overlap[NoLv,Lwv,L1v, RIn1v, hv, aCCv, epsVdWv]'";
"(Note: This function is omitted at calculations";
"    but could be helpful at calculation of VdW ebergy of monoscroll at debugging;";
"    for example,";
"    fEVdWLayer1Overlap[NoL1,Lwp,15nm, 2nm, hp, aCCp, epsVdWp] ";
" and fEVdWLayersOverlap[NoL2,Lwp,L1p=15nm, 2nm, hp, aCCp, epsVdWp, 0]";
"    give the same values";
fEVdWLayer1Overlap[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_, epsVdWv_] :=
   Module[
   {EVdWv,
     EVdW1Un1v = 0 (eV/atom), EVdW1Ov1v = 0 (eV/atom),
```

```
      Spirale1UnderSpirale1Length = 0 nm,
     Spirale1OverSpirale1Length = 0 nm,
      PhiIn1v = fPhiIn1[NoLv, RIn1v, hv],
      PhiOut1v = fPhiOut1[NoLv, L1v, RIn1v, hv],
      },
    Spirale1OverSpirale1Length =
     fSpiraleLen[NoLv, PhiIn1v + 2 Pi, PhiOut1v, hv];
    Spirale1UnderSpirale1Length =
     fSpiraleLen[NoLv, PhiIn1v , PhiOut1v - 2 Pi, hv];
    "Note: Spirale1OverSpirale1Length>Spirale1UnderSpirale1Length";
    EVdW1Un1v = -epsVdWv Lwv/(2 fSa[
          aCCv]) Spirale1UnderSpirale1Length;
    EVdW1Ov1v = -epsVdWv Lwv/(2 fSa[aCCv]) Spirale1OverSpirale1Length;
    EVdWv = (EVdW1Un1v + EVdW1Ov1v);
    Return[{EVdWv, EVdW1Un1v, EVdW1Ov1v}];
    ];
"The definition of the function";
"fEVdWLayersOverlap[NoLv_,Lwv_,L1v_, RIn1v_, hv_, aCCv_, epsVdWv_, dPhi12v_]";
fEVdWLayersOverlap[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_, epsVdWv_,
    dPhi12v_] := Module[
    {EVdW,
      EVdW1Un1 = 0 (eV/atom), EVdW1Ov1 = 0 (eV/atom),
      EVdW1Un2 = 0 (eV/atom), EVdW1Ov2 = 0 (eV/atom),
      EVdW2Un1 = 0 (eV/atom), EVdW2Ov1 = 0 (eV/atom),
      Spirale1UnderSpirale1Length = 0 nm,
     Spirale1OverSpirale1Length = 0 nm,
      Spirale1UnderSpirale2Length = 0 nm,
     Spirale1OverSpirale2Length = 0 nm,
      Spirale2UnderSpirale1Length = 0 nm,
     Spirale2OverSpirale1Length = 0 nm,
      PhiIn1 = fPhiIn1[NoLv, RIn1v, hv],
      PhiIn2 = fPhiIn2[NoLv, RIn1v, hv, dPhi12v],
      PhiOut1 = fPhiOut1[NoLv, L1v, RIn1v, hv],
      PhiOut2 = fPhiOut2[NoLv, L1v, RIn1v, hv, dPhi12v],
      ReturnEnergiesv = {1, 2, 3, 4, 5, 6, 7}
     },
    If[NoLv == 1,
     If[PhiIn1 < PhiOut1 - 2 Pi,
      Spirale1UnderSpirale1Length =
        fSpiraleLen[NoLv, PhiIn1, PhiOut1 - 2 Pi, hv];];
     If[PhiIn1 + 2 Pi < PhiOut1,
      Spirale1OverSpirale1Length =
        fSpiraleLen[NoLv, PhiIn1 + 2 Pi, PhiOut1, hv];];
     EVdW1Un1 = -epsVdWv Lwv/(2 fSa[
           aCCv]) Spirale1UnderSpirale1Length;
     EVdW1Ov1 = -epsVdWv Lwv/(2 fSa[aCCv]) Spirale1OverSpirale1Length;
     EVdW = (EVdW1Un1 + EVdW1Ov1);
     ReturnEnergiesv = {EVdW, EVdW1Un1, EVdW1Ov1};
     ];
    If[NoLv == 2,
        If[PhiIn1 < PhiOut2 - Pi,
      Spirale1UnderSpirale2Length =
        fSpiraleLen[NoLv, PhiIn1, PhiOut2 - Pi, hv];];
         If[PhiIn1 + Pi + dPhi12v < PhiOut1,
      Spirale1OverSpirale2Length =
        fSpiraleLen[NoLv, PhiIn1 + Pi + dPhi12v, PhiOut1, hv];];
         If[PhiIn1 + dPhi12v < PhiOut1 - Pi,
```

```
      Spirale2UnderSpirale1Length =
        fSpiraleLen[NoLv, PhiIn1 + dPhi12v, PhiOut1 - Pi, hv];];
         If[PhiIn1 - dPhi12v + Pi < PhiOut2 - dPhi12v,
      Spirale2OverSpirale1Length =
        fSpiraleLen[NoLv, PhiIn1 - dPhi12v + Pi, PhiOut2 - dPhi12v,
         hv];];
     EVdW1Un2 = -epsVdWv Lwv/(2 fSa[
           aCCv]) Spirale1UnderSpirale2Length;
          EVdW1Ov2 = -epsVdWv Lwv/(2 fSa[
           aCCv]) Spirale1OverSpirale2Length;
          EVdW2Un1 = -epsVdWv Lwv/(2 fSa[
           aCCv]) Spirale2UnderSpirale1Length;
          EVdW2Ov1 = -epsVdWv Lwv/(2 fSa[
           aCCv]) Spirale2OverSpirale1Length;
          EVdW = (EVdW1Un2 + EVdW1Ov2 + EVdW2Un1 + EVdW2Ov1);
          ReturnEnergiesv[[iEVdW]] = EVdW;
          ReturnEnergiesv[[iEVdW1Un2]] = EVdW1Un2;
          ReturnEnergiesv[[iEVdW1Ov2]] = EVdW1Ov2;
          ReturnEnergiesv[[iEVdW2Un1]] = EVdW2Un1;
          ReturnEnergiesv[[iEVdW2Ov1]] = EVdW2Ov1;
     ];
    Return[ReturnEnergiesv];
    ];

EVdWdPhi12eq0allp =
    fEVdWLayersOverlap[NoL2, Lwp, L1p, RIn1p, hp, aCCp, epsVdWp,
   dPhi12eq0];
EVdWvardPhi12allp =
    fEVdWLayersOverlap[NoLp, Lwp, L1p, RIn1p, hp, aCCp, epsVdWp,
   dPhi12p];
If[NoLp == 1,
  Print[" EVdWvardPhi12allp[[iEVdW]]=",
   EVdWvardPhi12allp[[iEVdW]]/(eV/atom), "eV/atom"];
  Print["( EVdWvardPhi12allp[[iEVdW1Un1]]=",
   EVdWvardPhi12allp[[iEVdW1Un1]]/(eV/atom), "eV/atom"];
    Print["  EVdWvardPhi12allp[[iEVdW1Ov1]]=",
   EVdWvardPhi12allp[[iEVdW1Ov1]]/(eV/atom), "eV/atom  )"];
  ];
If[NoLp == 2,
  Print[" for dPhi12=", dPhi12p/Pi,
   "Pi EVdWvardPhi12allp[[iEVdW]]=",
     EVdWvardPhi12allp[[iEVdW]]/(eV/atom), "eV/atom"];
    Print[" For dPhi12=", dPhi12eq0/Pi, "Pi:"];
    Print[" EVdWvardPhi12allp[[iEVdW]]=",
      EVdWdPhi12eq0allp[[iEVdW]]/(eV/atom), "eV/atom"];
    Print[" EVdWvardPhi12allp[[iEVdW1Un2]]=",
      EVdWdPhi12eq0allp[[iEVdW1Un2]]/(eV/atom), "eV/atom"];
    Print[" EVdWvardPhi12allp[[iEVdW1Ov2]]=",
      EVdWdPhi12eq0allp[[iEVdW1Ov2]]/(eV/atom), "eV/atom"];
    Print[" EVdWvardPhi12allp[[iEVdW2Un1]]=",
      EVdWdPhi12eq0allp[[iEVdW2Un1]]/(eV/atom), "eV/atom"];
    Print[" EVdWvardPhi12allp[[iEVdW2Ov2]]=",
      EVdWdPhi12eq0allp[[iEVdW2Ov1]]/(eV/atom), "eV/atom"];
    EVdWdPhi12eqPiallp =
      fEVdWLayersOverlap[NoLp, Lwp, L1p, RIn1p, hp, aCCp, epsVdWp,
    dPhi12eqPi];
  (**)  Print[" For dPhi12=", dPhi12eqPi/Pi, "Pi:"];
```

```
     Print[" EVdWvatdPhi12allp[[iEVdW]]=",
       EVdWdPhi12eqPiallp[[iEVdW]]/(eV/atom), "eV/atom"];
     Print[" EVdWvatdPhi12allp[[iEVdW1Un2]]=",
       EVdWdPhi12eqPiallp[[iEVdW1Un2]]/(eV/atom), "eV/atom"];
     Print[" EVdWvatdPhi12allp[[iEVdW1Ov2]]=",
       EVdWdPhi12eqPiallp[[iEVdW1Ov2]]/(eV/atom), "eV/atom"];
     Print[" EVdWvatdPhi12allp[[iEVdW2Un1]]=",
       EVdWdPhi12eqPiallp[[iEVdW2Un1]]/(eV/atom), "eV/atom"];
     Print[" EVdWvatdPhi12allp[[iEVdW2Ov2]]=",
       EVdWdPhi12eqPiallp[[iEVdW2Ov1]]/(eV/atom), "eV/atom"];
     EVdWEVdWdPhi12eq0p = EVdWdPhi12eq0allp[[iEVdW]];
     Print[" EVdWdPhi12eq0allp=", EVdWdPhi12eq0allp/(eV/atom),
       "eV/atom"];
     EVdWEVdWdPhi12eqPip = EVdWvardPhi12allp[[iEVdW]];
     Print[" EVdWEVdWdPhi12eqPip=", EVdWEVdWdPhi12eqPip/(eV/atom),
       "eV/atom"];
   (**)
   ];
If[NoLp == 2, Print[" III.3. The energy of flat planes "];];
fEnergyFlatPlanes[NoLv_, Lwv_, L1v_, aCCv_, epsVdWv_] :=
  If[NoLv == 2, -epsVdWv Lwv/fSa[aCCv] L1v, 0 eV/atom];
EnergyFlatPlanesp = fEnergyFlatPlanes[NoLp, Lwp, L1p, aCCp, epsVdWp];
If[NoLp == 2, Print[" EnergyFlatPlanes=-eps width/Sa L1(NoL-1) =",
      EnergyFlatPlanesp/(eV/atom), "eV/atom"];];
Print[" III.4. The total energy of the nanoscroll"];
fScrollEnergydPhi[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_, epsVdWv_,
   CCv_, CBNv_, dPhi12v_] :=
   Module[{ScrollEnergyv, EVdWv, EVdWnoDimv},
     EVdWv =
    fEVdWLayersOverlap[NoLv, Lwv, L1v, RIn1v, hv, aCCv, epsVdWv,
       dPhi12v][[1]];
   EVdWnoDimv = EVdWv /. {eV -> 1, atom -> 1, nm -> 1};
     If[NoLv == 1,

    If[EVdWnoDimv == 0,
       ScrollEnergyv =
        fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv],

       ScrollEnergyv =
         (EVdWv/(eV/atom) + fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv]/(eV/atom))(eV/at
         ];
     If[NoLv == 2,
    If[EVdWnoDimv == 0,
       ScrollEnergyv =
        fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv] +
                 fEelastCBN[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv],
       ScrollEnergyv =
         (EVdWv + fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv]/(eV/atom) +
                  fEelastCBN[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv]/(eV/atom))(eV/atom);];
      ];
     Return[ScrollEnergyv];
   ];

fScrollEnergyVdWandElast[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_,
   epsVdWv_, CCv_, CBNv_] :=
   Module[{ ScrollEnergyVdWandElastv, EVdWv},
   (*If[NoLv == 1,
```

```
     EVdWv=fEVdWLayer1Overlap[NoLv,Lwv,L1v,RIn1v,hv,aCCv,
    epsVdWv][[1]];
    ];
    If[NoLv == 2,
     EVdWv=fEVdWLayersOverlap[NoLv,Lwv,L1v,RIn1v,hv,aCCv,
    epsVdWv][[1]];
    ];*)
    EVdWv =
     fEVdWLayer1Overlap[NoLv, Lwv, L1v, RIn1v, hv, aCCv,
       epsVdWv][[1]];
    If[NoLv == 1,
       ScrollEnergyVdWandElastv =
       EVdWv + fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv];];
    If[NoLv == 2,
       ScrollEnergyVdWandElastv =
       EVdWv + fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv] +
        fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CBNv];];

    Return[ScrollEnergyVdWandElastv];
    ];

fScrollEnergy[NoLv_, Lwv_, L1v_, RIn1v_, hv_, aCCv_, epsVdWv_, CCv_,
    CBNv_] := Module[
    {ScrollEnergyv = -10^20 eV/atom},
     If[RIn1v/m <= fRIn1Sharp[NoLv, L1v, hv]/m,
     ScrollEnergyv =
      fScrollEnergyVdWandElast[NoLv, Lwv, L1v, RIn1v, hv, aCCv,
       epsVdWv, CCv, CBNv];
     "note: the function fScrollEnergyVdWandElast[1,..] is analytycal";
     "whereas the function fScrollEnergy[....] uses the 'If[..]'- function";
     ];
     If[RIn1v/m >= fRIn1Sharp[NoLv, L1v, hv]/m,
     If[NoLv == 1,
      ScrollEnergyv =
        fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv];];
     If[NoLv == 2,
      ScrollEnergyv =
        fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CCv] +
         fEelastCC[NoLv, Lwv, L1v, RIn1v, hv, aCCv, CBNv];];

     ];
    Return[ScrollEnergyv];
    ];

ScrollEnergyp =
  fScrollEnergy[NoLp, Lwp, L1p, RIn1p, hp, aCCp, epsVdWp, CCp,
    CBNp];
ScrollEnergyp =
  fScrollEnergy[1, Lwp, L1p, RIn1p, hp, aCCp, epsVdWp, CCp, CBNp];
Print["fScrollEnergy[1,Lwp,L1p, RIn1p,hp, aCCp, epsVdWp,CCp,CBNp]="];
Print["=fScrollEnergy[1, Lw=", Lwp/nm, "nm, L1=", L1p/nm, "nm, RIn1=",
    RIn1p/nm, "nm, h=", hp/nm, "nm,"];
Print[" aCC=", aCCp/nm, "nm, epsVdW=", epsVdWp/(eV/atom),
  "eV/atom, CC=", CCp/(eV AA^2/atom), "(eV AA^2/atom)="];
Print["            =", ScrollEnergyp/(eV/atom), "eV/atom"];

Print[" III.5. Determine the inner angles mismatch for the bi-layer nanoscroll
```

```
        for the high nanoribbon Length"];
Print[" For L1=", L1p/nm, "nm,RIn=", RIn1p/nm, "nm,h=", hp/nm,
  "nm  and  dPhi12=0:"];
Print[" ScrollEnergy=", ScrollEnergyp/(eV/atom), "eV/atom"];
Print[" For L1=", L1p/nm, "nm,RIn=", RIn1p/nm, "nm,h=", hp/nm,
  "nm  and  dPhi12=Pi:"];
Print[" ScrollEnergy=", ScrollEnergyp/(eV/atom), "eV/atom"];
Print["-------------------------------------------------------------"];

Print[" IV.The potential energy of the nanoscroll"];
Print[" as a function of the inner radius RIn"];
Print[" NoL=", NoLp];
Print[" epsVdW=", epsVdWp/(eV/atom), "eV/atom, C=",
  CCp/(eV nm^2/atom), "(eV nm^2/atom)",
    "(eV nm^2/atom),aCC=", aCCp/nm, "nm,h=", hp/nm, "nm"];
Print[" Plot ScrollEnergy[RIn1/nm]/(eV/atom) for L1=", L1p/nm,
    "nm (NoL=", NoLp, ",Lw=", Lwp/nm, "nm"];
PlotScrollEnergyVsRIn1 =
  Plot[(fScrollEnergy[NoLp, Lwp, L1p, RIn1nmv nm, hp, aCCp, epsVdWp,
            CCp, CBNp])/(eV/
            atom), {RIn1nmv, RIn1Minp/nm, RIn1Maxp/nm},
   PlotRange -> PlotRangep/(eV/atom)];
Print[PlotScrollEnergyVsRIn1];
Print[" Plot ScrollEnergy[RIn1/nm]/(eV/atom) for L1=", tL1/nm,
    "nm (NoL=", NoLp, ",w=", Lwp/nm, "nm)"];
PlotScrollEnergyVsRIn1L1th =
    Plot[(fScrollEnergy[NoLp, Lwp, tL1[[1]], RIn1nmv nm, hp, aCCp,
      epsVdWp,
            CCp, CBNp] )/(eV/
            atom), {RIn1nmv, RIn1Minp/nm, RIn1Maxp/nm},
   PlotRange -> PlotRangep/(eV/atom)];
PlotScrollEnergyVsRIn1L2th =
    Plot[(fScrollEnergy[NoLp, Lwp, tL1[[2]], RIn1nmv nm, hp, aCCp,
      epsVdWp,
            CCp, CBNp] )/(eV/
            atom), {RIn1nmv, RIn1Minp/nm, RIn1Maxp/nm},
   PlotRange -> PlotRangep/(eV/atom)];
PlotScrollEnergyVsRIn1L3th =
    Plot[(fScrollEnergy[NoLp, Lwp, tL1[[3]], RIn1nmv nm, hp, aCCp,
      epsVdWp,
            CCp, CBNp] )/(eV/
            atom), {RIn1nmv, RIn1Minp/nm, RIn1Maxp/nm},
   PlotRange -> PlotRangep/(eV/atom)];
PlotScrollEnergyVsRIn1L4th =
    Plot[(fScrollEnergy[NoLp, Lwp, tL1[[4]], RIn1nmv nm, hp, aCCp,
      epsVdWp,
            CCp, CBNp])/(eV/
            atom), {RIn1nmv, RIn1Minp/nm, RIn1Maxp/nm},
   PlotRange -> PlotRangep/(eV/atom)];
Print[Show[{PlotScrollEnergyVsRIn1L1th, PlotScrollEnergyVsRIn1L2th,
        PlotScrollEnergyVsRIn1L3th, PlotScrollEnergyVsRIn1L4th}]];
Print["The examples of using of 'fScrollEnergy[..]' function:"]
Print["fScrollEnergy[NoLp,Lwp,tL1[[1]],RIn1p,hp, aCCp, epsVdWp,CCp,CBNp]=", fScrollEnergy[
    CCp, CBNp]/(eV/atom), " eV/atom"];
Print["fScrollEnergy[NoLp,Lwp,tL1[[1]],1nm,hp, aCCp, epsVdWp,CCp, CBNp]=", fScrollEnergy[N
    CCp, CBNp]/(eV/atom), " eV/atom"];
Print["fScrollEnergy[NoLp,Lwp,7nm,1nm,hp, aCCp, epsVdWp,CCp, CBNp]=",
```

```
  fScrollEnergy[NoLp, Lwp, 7. nm, 1. nm, hp, aCCp, epsVdWp, CCp,
    CBNp]/(eV/atom), " eV/atom"];
Print["fEVdWLayer1Overlap[1,Lwp,7.nm,1.nm,hp,aCCp,epsVdWp][[1]]=",
  fEVdWLayer1Overlap[1, Lwp, 7. nm, 1. nm, hp, aCCp,
     epsVdWp][[1]]/(eV/atom),
  "eV/atom (right !=0 value, because the layer overlaps"];
Print["fEVdWLayer1Overlap[1,Lwp,7.nm,1.5nm,hp,aCCp,epsVdWp][[1]]=",
  fEVdWLayer1Overlap[1, Lwp, 7. nm, 2.5 nm, hp, aCCp,
     epsVdWp][[1]]/(eV/atom),
  "eV/atom !=0, wrong value of the fEVdWLayer1Overlap[..] function because the layer does
Print[];
Print["The analytical expressions of the fEVdWLayer1Overlap[..] function:"];
Print[];
Print["--- fEVdWLayer1Overlap[NoL1,Lwv,Lpv,RInv,hv,aCCv,epsVdWv][[1]]: ---"];
Print[fEVdWLayer1Overlap[NoL1, Lwv, Lpv, RInv, hv, aCCv,
    epsVdWv][[1]]];
Print[];
Print["--- fEVdWLayer1Overlap[NoL2,Lwv,Lpv,RInv,hv,aCCv,epsVdWv][[1]]: ---"];
Print[fEVdWLayer1Overlap[NoL2, Lwv, Lpv, RInv, hv, aCCv,
    epsVdWv][[1]]];
Print[];
Print["The Analytical expression of "];
Print["fScrollEnergyVdWandElast[", NoLp,
  ",Lwv,L1v,RIn1v,hv,aCCv, epsVdWv, CCv,CBNv]:"];
Print[fScrollEnergyVdWandElast[NoLp, Lwv, L1v, RIn1v, hv, aCCv,
   epsVdWv, CCv, CBNv]];
Print["-----------------------------------------------------------"];

Print["V. Export the data of the plots of the nanoscroll energy"];
Print["   as a function of the inner radius"];
Print["The parameters of the output file"];
Print["The number of the output points = ", npRIn1];
Print["Export the plot data to the files:"];
tRIn1nmRegular =
  Table[(RIn1Minp + (RIn1Maxp - RIn1Minp) iiRin/(npRIn1))/nm, {iiRin,
    1, npRIn1}];
tScrollEnergy = tRIn1nmRegular;
tScrollEnergyeVatom = tRIn1nmRegular;
tPlotEvsRin = Table[{}, {ii, 1, Length[tL1]}];
AllPlotsEVsRin = {};

For[iiL1 = 1, iiL1 <= Length[tL1], iiL1++,
  L1pi = tL1[[iiL1]];
  NanoscrollNamep =
   StringJoin["Nanoscroll", ToString[NoLp], "L", ToString[L1pi/nm],
    "nm"];
  Print["NanoscrollName=", NanoscrollNamep];
  ScrollEnergyFileName =
   StringJoin["EvsRIn1", NanoscrollNamep, ".dat"];
  Print["ScrollEnergyFileName=", ScrollEnergyFileName];

  For[iiRIn1 = 1, iiRIn1 <= npRIn1, iiRIn1++,
   RIn1pi = tRIn1nmRegular[[iiRIn1]] nm;
   tScrollEnergy[[iiRIn1]] =
    fScrollEnergy[NoLp, Lwp, L1pi, RIn1pi, hp, aCCp, epsVdWp,
         CCp, CBNp];
```

```
        tScrollEnergyeVatom[[iiRIn1]] = (tScrollEnergy[[iiRIn1]])/(eV/atom);];
       tPlotEvsRin[[iiL1]] =
        ListPlot[Transpose[{tRIn1nmRegular, tScrollEnergyeVatom}],
         PlotRange -> PlotRangep/(eV/atom)];
       Print[tPlotEvsRin[[iiL1]]];
       AllPlotsEVsRin = Join[{AllPlotsEVsRin, tPlotEvsRin[[iiL1]]}];
       CarbonNanoscrollEnergyVsRinFileName =
        StringJoin[NanoscrollNamep, "dat"];
       Export[ToFileName[NotebookDirectory[], ScrollEnergyFileName],
        Transpose[{Insert[tRIn1nmRegular, "RIn1[nm]", 1],
          Insert[tScrollEnergyeVatom, "E[eV/atom]", 1]}]]
       ];
     Print["Plot ScrollEnergy[RIn1/nm]/(eV/atom) for L1=", tL1/nm,
         "nm (NoL=", NoLp, ",Lw=", Lwp/nm, "nm)"];
     Print[Show[AllPlotsEVsRin]];

     Print[];
     Print["Manipulating of the plot of the nanoscroll energy as the functoon"];
     Print[" of the nanoscroll inner radius:"];
     Print[" "];
     Print["(Manipulate[Plot[fScrollEnergy[...,L1nmm,RIn1,hp,...]]],"];
     Print["                       where L1nmm (is L1 in nanometers) "];
     Print["                       is the manipulated value"];
     Print["                      )"];
     Manipulate[
      Plot[(fScrollEnergy[NoLp, Lwp, L1nmm nm, RIn1nmv nm, hp, aCCp,
          epsVdWp,
                        CCp, CBNp])/(eV/atom), {RIn1nmv, RIn1Minp/nm,
        RIn1Maxp/nm}, PlotRange -> PlotRangep/(eV/atom)], {{L1nmm, L1p/nm},
        0.5 tL1[[1]]/nm, 1.5 tL1[[Length[tL1]]]/nm}]
```

```
 Figure3cd(Layers1L15nmRIn1d14.m, Figure3cd(Layers2L15nmRIn1d14.m

  The source of the data of the manuscript
  'Structure and energetics of carbon,
  hexagonal boron nitride, and
  carbon/hexagonal boron nitride
  single-layer and bilayer nanoscrolls'
/ A.I. Siahlo, N.A. Poklonski, A.V. Lebedev,
I.V. Lebedeva, A.M. Popov, S.A. Vyrko,
A.A. Knizhnik, Yu.E. Lozovik
// Phys. Rev. Materials.— 2018.— V. 2,
№ 3.— P. 036001 (9 pp.).
[DOI: 10.1103/PhysRevMaterials.2.036001]
---------------------------------------------------
I.0 The Units (nm, meV, AA)
-----------------------------------------------------------
```

```
 I.1. All Input Parameters and Constants--------------
 I.1.1. The sampling parameters
npRIn1=1000
 I.1.2. The Input Geometry Parameters of the system
 The carbon nanoribbon length L1=14.839nm
 The carbon nanoribbon width Lw=11.8nm
-----------------------------------------------------------
 Number of the layers in carbon nanoscroll NoL=1
 The length of a carbon nanoribbon L1=14.839nm
 The carbon nanoribbon width Lw=1.nm
 The inner radius of the nanoscroll RIn1=1.14nm
 I.1.2. The Input Energy Constants
 epsVdW - the interlayer interaction energy per one atom of
 the nanoscroll:
 epsVdW=0.035eV/atom
 C - the bending elastic constant:
 CCelast=2.01eV AA^2/atom
 CCBNelast=2.01eV AA^2/atom
 I.1.3.The Input Geometry constants--------------
 The interatomic distance aCC and the interlayer distance h
h=0.3354 nm  (Interlayer distance)
 aCC=0.142nm, h=0.3354nm
NatomsInCell=2
 dPhi12 - The difference of the inner angles of the spirales
   of the Layers
 I.4.The parameters for the visualisation
 I.5. The parameters of visualization that depend on NoL=1
 I.6. The parameters of the output file
 NanoscrollName=Nanoscroll1L14.839nm
Nanoscroll1L14.839nm.txt
 (The output of the data to a file Is Not Performed)
 The number of the output points = 1000
 I.7. The Input Numerical Constants used in the programm
 The Indexes used for the work with EVdW[...] function
 --------End of the Input---------------
-----------------------------------------------------------
 II. The derivated parameters and the functions required
```

II.1. The derivated parameters

The cell area Sa= $\frac{3\sqrt{3}\ aCC^2}{4}$ =0.0261938nm^2

II.2. The required functions--------------

II.2.1. The function fSpiraleLen[1,φIn, φOut, h] defines

the Length of a Spirale with the inner agle φIn and the outer angle φOut(>=φIn):

$$fSpiraleLen[1,\ \varphi In,\ \varphi Out,\ h] = \frac{h\left(-\varphi In\sqrt{1+\varphi In^2}+\varphi Out\sqrt{1+\varphi Out^2}-ArcSinh[\varphi In]+ArcSinh[\varphi Out]\right)}{4\pi}.$$

II.2.2. The function fElast[φIn,ρOut] is required to calculate an nanoscrollelastic energy:

$$fElast[\varphi In,\varphi Out] = \frac{\sqrt{1+\varphi In^2}}{\varphi In}-\frac{\sqrt{1+\varphi Out^2}}{\varphi Out}-ArcSinh[\varphi In]+ArcSinh[\varphi Out].$$

II.2.3. Define the function fPhiOutvsPhiInLh[1,φIn,L,h].

The function fPhiOutvsPhiInLh[1,φIn,L,h]= $\sqrt{\frac{4L\pi}{h}+\varphi In^2}$ is a

good approximation to obtain the value of φOut for the defined φIn,L,h.

The inverse function fPhiInvsPhiOutLh[1,φOut, L, h]]= $\sqrt{-\frac{4L\pi}{h}+\varphi Out^2}$

could be used in the program applications
if ROut (instead of RIn) is the input parameter of the system.

II.2.4. The functions fSpirale1Under(Over)Spirale1Length[NoLv,PhiIn1v ,PhiOut1v,hv]

These functiona are not required, but could be helpful),

fSpirale1UnderSpirale1Length[1,PhiIn1v ,PhiOut1v,hv]=

=fSpiraleLen[NoLv,PhiIn1v ,PhiOut1v-2Pi,hv]= $\frac{1}{4\pi}$

$$hv\ NoLv\left(-PhiIn1v\sqrt{1+PhiIn1v^2}+\sqrt{1+(PhiOut1v-2\pi)^2}\ (PhiOut1v-2\pi)-ArcSinh[PhiIn1v]+ArcSinh[PhiOut1v-2\pi]\right);$$

fSpirale1UnderSpirale1Length[NoLp,PhiIn1p,PhiOut1p,hp]=fSpirale1UnderSpirale1Length[
1,3.39893(2Pi),5.06316(2Pi),0.3354nm] =

=5.22745nm.

fSpirale1OverSpirale1Length[1,PhiIn1v ,PhiOut1v,hv]=

=fSpiraleLen[NoLv,PhiIn1v+2Pi ,PhiOut1v,hv]= $\frac{1}{4\pi}$

$$hv\ NoLv\left(PhiOut1v\sqrt{1+PhiOut1v^2}+(-PhiIn1v-2\pi)\sqrt{1+(PhiIn1v+2\pi)^2}+ArcSinh[PhiOut1v]-ArcSinh[PhiIn1v+2\pi]\right);$$

fSpirale1OverSpirale1Length[NoLp,PhiIn1p ,PhiOut1p,hp]=fSpirale1OverSpirale1Length[
1,3.39893(2Pi),5.06316(2Pi),0.3354nm] =

=6.62623nm.

II.2.4. The function fRIn1Sharp[NoLv,L1v,hv]

$$\text{fRIn1Sharp[NoLv,L1v,hv]} = -\frac{\text{hv NoLv}}{2} + \frac{\text{L1v}}{2\pi}$$

is a good approximation to obtain the value of the sharp in the dependence ScrollEnergy[RIn]

fRIn1Sharp[1, 14.839nm, 0.3354nm] = 2.194nm

-----------------------------------------------------------

III. Begin of Calculation

III.1. The inner and the outer angle of the spirale of the layer:

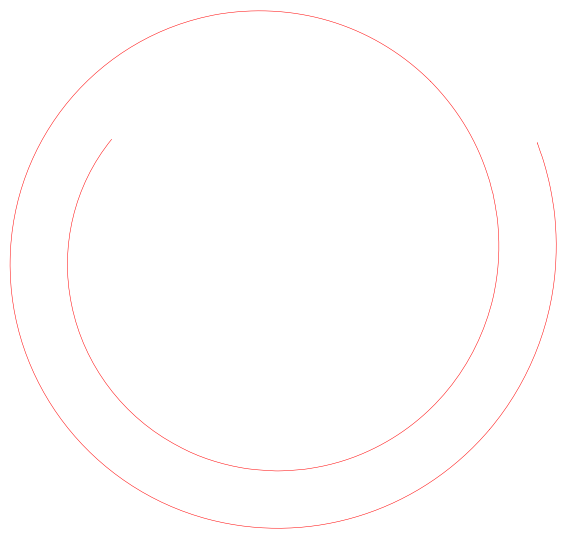
$\varphi\text{In1} = \frac{2\pi\,\text{RIn1}}{\text{h}}$, $\varphi$Out1=fPhiOutvsPhiInLh[1,$\phi$In1,L1,h];

For RIn1=1.14nm,h=0.3354nm:

$\varphi$In1=3.39893(2Pi),$\Phi$Out1=5.06316(2Pi).

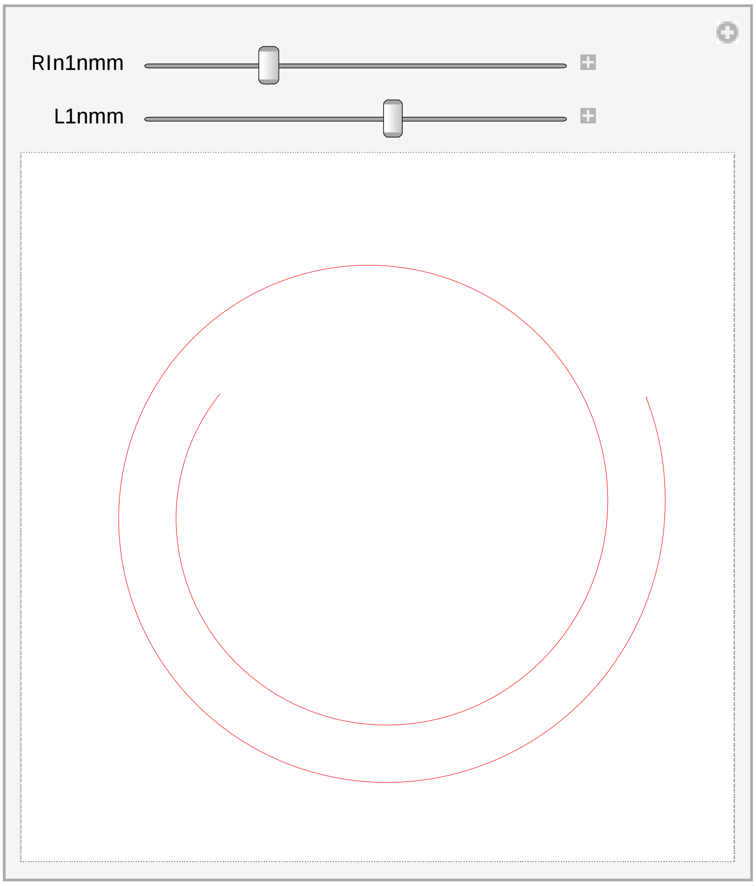
L1=14.839nm, RIn1=1.14nm

Plot the Spirale of the layer:

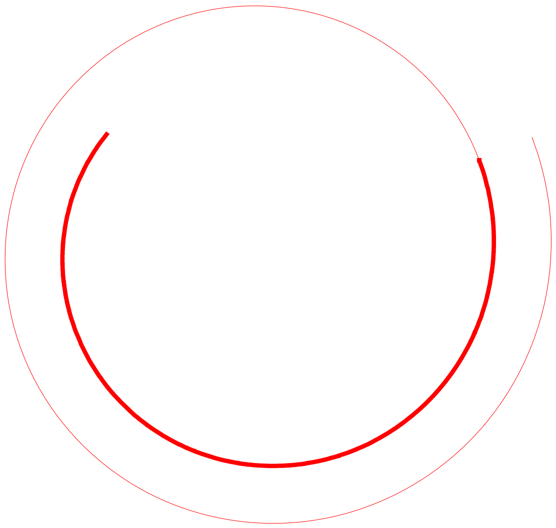


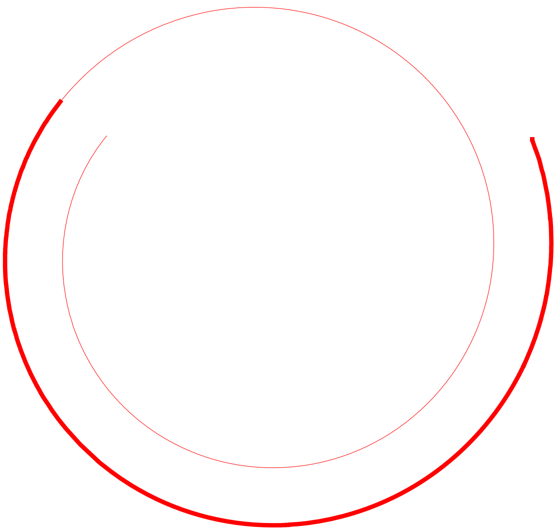
Manipulating of Spirale1Plot for the different RIn1 and L:

Out[454]=




```
{Spirale,Spirale1UnderSpirale1},{Spirale1,Spirale1OverSpirale1}:
```





III.2. The nanoscroll energy calculation

III.2.1. The elastic energy calculation

EelastC=5.7333eV/atom

III.2.2. The Van-der-Waals energy calculation

EVdWvardPhi12allp[[iEVdW]]=-7.91941eV/atom

( EVdWvardPhi12allp[[iEVdW1Un1]]=-3.49244eV/atom

EVdWvardPhi12allp[[iEVdW1Ov1]]=-4.42696eV/atom )

III.4. The total energy of the nanoscroll

fScrollEnergy[1,Lwp,L1p, RIn1p,hp, aCCp, epsVdWp,CCp,CBNp]=

=fScrollEnergy[1, Lw=1.nm, L1=14.839nm, RIn1=1.14nm, h=0.3354nm, aCC=0.142nm, epsVdW=0.035eV/atom, CC=2.01(eV AA^2/atom)=

=-2.18611eV/atom

III.5. Determine the inner angles mismatch for the bi-layer nanoscroll
for the high nanoribbon Length

For L1=14.839nm,RIn=1.14nm,h=0.3354nm and dPhi12=0:

ScrollEnergy=-2.18611eV/atom

For L1=14.839nm,RIn=1.14nm,h=0.3354nm and dPhi12=Pi:

ScrollEnergy=-2.18611eV/atom

-------------------------------------------------------------

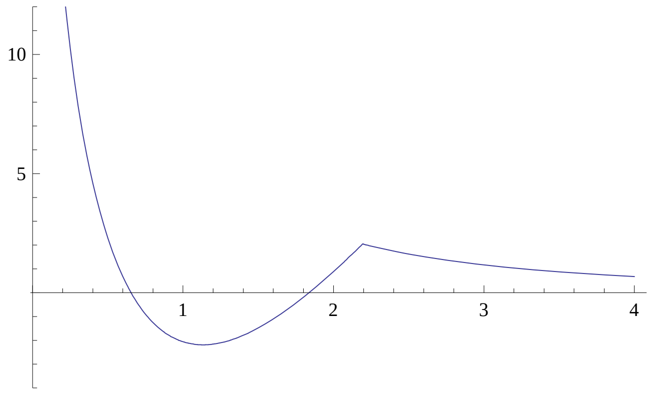
IV.The potential energy of the nanoscroll

as a function of the inner radius RIn

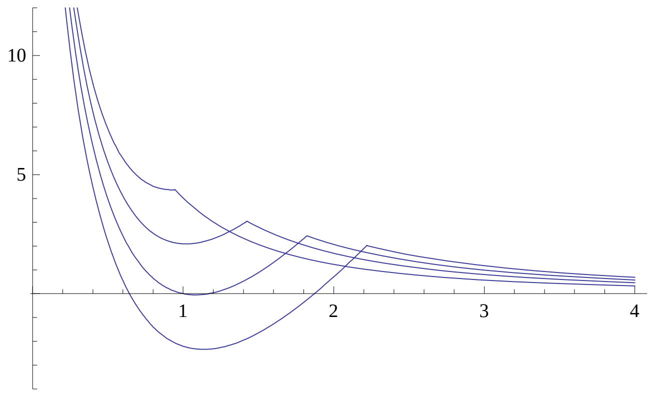
NoL=1

epsVdW=0.035eV/atom, C=0.0201(eV nm^2/atom)(eV nm^2/atom),aCC=0.142nm,h=0.3354nm

Plot ScrollEnergy[RIn1/nm]/(eV/atom) for L1=14.839nm (NoL=1,Lw=1.nm




Plot ScrollEnergy[RIn1/nm]/(eV/atom) for L1={7., 10., 12.5, 15.}nm (NoL=1,w=1.nm)




The examples of using of 'fScrollEnergy[..]' function:

fScrollEnergy[NoLp,Lwp,tL1[[1]],RIn1p,hp, aCCp, epsVdWp,CCp,CBNp]=3.26809 eV/atom

fScrollEnergy[NoLp,Lwp,tL1[[1]],1nm,hp, aCCp, epsVdWp,CCp, CBNp]=4.01569 eV/atom

fScrollEnergy[NoLp,Lwp,7nm,1nm,hp, aCCp, epsVdWp,CCp, CBNp]=4.01569 eV/atom

fEVdWLayer1Overlap[1,Lwp,7.nm,1.nm,hp,aCCp,epsVdWp][[1]]=
0.39374eV/atom (right !=0 value, because the layer overlaps

```
fEVdWLayer1Overlap[1,Lwp,7.nm,1.5nm,hp,aCCp,epsVdWp][[1]]=
 12.2479eV/atom !=0, wrong value of the
   fEVdWLayer1Overlap[..] function because the layer does not not overlap
```

```
The analytical expressions of the fEVdWLayer1Overlap[..] function:
```

```
--- fEVdWLayer1Overlap[NoL1,Lwv,Lpv,RInv,hv,aCCv,epsVdWv][[1]]: ---
```

$$-\frac{1}{6\sqrt{3}\ \text{aCCv}^2\,\pi}\text{epsVdWv hv Lwv}\left(\sqrt{\frac{4\,\text{Lpv}\,\pi}{\text{hv}}+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}\ \sqrt{1+\frac{4\,\text{Lpv}\,\pi}{\text{hv}}+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}+\left(-2\,\pi-\frac{2\,\pi\,\text{RInv}}{\text{hv}}\right)\sqrt{1+\left(2\,\pi+\frac{2\,\pi\,\text{RInv}}{\text{hv}}\right)^2}-\text{ArcSinh}\left[2\,\pi+\frac{2\,\pi\,\text{RInv}}{\text{hv}}\right]+\text{ArcSinh}\left[\sqrt{\frac{4\,\text{Lpv}\,\pi}{\text{hv}}+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right]\right)-\frac{1}{6\sqrt{3}\ \text{aCCv}^2\,\pi}\text{epsVdWv hv Lwv}\left(-\frac{2\,\pi\,\text{RInv}\sqrt{1+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}}{\text{hv}}+\left(-2\,\pi+\sqrt{\frac{4\,\text{Lpv}\,\pi}{\text{hv}}+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right)\sqrt{1+\left(-2\,\pi+\sqrt{\frac{4\,\text{Lpv}\,\pi}{\text{hv}}+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right)^2}-\text{ArcSinh}\left[\frac{2\,\pi\,\text{RInv}}{\text{hv}}\right]-\text{ArcSinh}\left[2\,\pi-\sqrt{\frac{4\,\text{Lpv}\,\pi}{\text{hv}}+\frac{4\,\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right]\right)$$

```
--- fEVdWLayer1Overlap[NoL2,Lwv,Lpv,RInv,hv,aCCv,epsVdWv][[1]]: ---
```

$$-\frac{1}{3\sqrt{3}\ \text{aCCv}^2\,\pi}\text{epsVdWv hv Lwv}\left(\sqrt{\frac{2\,\text{Lpv}\,\pi}{\text{hv}}+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}\ \sqrt{1+\frac{2\,\text{Lpv}\,\pi}{\text{hv}}+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}+\left(-2\,\pi-\frac{\pi\,\text{RInv}}{\text{hv}}\right)\sqrt{1+\left(2\,\pi+\frac{\pi\,\text{RInv}}{\text{hv}}\right)^2}-\text{ArcSinh}\left[2\,\pi+\frac{\pi\,\text{RInv}}{\text{hv}}\right]+\text{ArcSinh}\left[\sqrt{\frac{2\,\text{Lpv}\,\pi}{\text{hv}}+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right]\right)-\frac{1}{3\sqrt{3}\ \text{aCCv}^2\,\pi}\text{epsVdWv hv Lwv}\left(-\frac{\pi\,\text{RInv}\sqrt{1+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}}{\text{hv}}+\left(-2\,\pi+\sqrt{\frac{2\,\text{Lpv}\,\pi}{\text{hv}}+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right)\sqrt{1+\left(-2\,\pi+\sqrt{\frac{2\,\text{Lpv}\,\pi}{\text{hv}}+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right)^2}-\text{ArcSinh}\left[\frac{\pi\,\text{RInv}}{\text{hv}}\right]-\text{ArcSinh}\left[2\,\pi-\sqrt{\frac{2\,\text{Lpv}\,\pi}{\text{hv}}+\frac{\pi^2\,\text{RInv}^2}{\text{hv}^2}}\right]\right)$$

The Analytical expression of

fScrollEnergyVdWandElast[1,Lwv,L1v,RIn1v,hv,aCCv, epsVdWv, CCv,CBNv]:

$$\frac{8\,\text{CCv Lwv}\,\pi\left(\frac{\text{hv}\sqrt{1+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}}{2\pi\,\text{RIn1v}}-\frac{\sqrt{1+\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}}{\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}}-\text{ArcSinh}\left[\frac{2\pi\,\text{RIn1v}}{\text{hv}}\right]+\text{ArcSinh}\left[\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}\right]\right)}{3\sqrt{3}\,\text{aCCv}^2\,\text{hv}}-$$

$$\frac{1}{6\sqrt{3}\,\text{aCCv}^2\,\pi}\text{epsVdWv hv Lwv}\left(\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}\sqrt{1+\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}+\left(-2\pi-\frac{2\pi\,\text{RIn1v}}{\text{hv}}\right)\sqrt{1+\left(2\pi+\frac{2\pi\,\text{RIn1v}}{\text{hv}}\right)^2}-\right.$$

$$\left.\text{ArcSinh}\left[2\pi+\frac{2\pi\,\text{RIn1v}}{\text{hv}}\right]+\text{ArcSinh}\left[\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}\right]\right)-\frac{1}{6\sqrt{3}\,\text{aCCv}^2\,\pi}\text{epsVdWv hv Lwv}$$

$$\left(-\frac{2\pi\,\text{RIn1v}\sqrt{1+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}}{\text{hv}}+\left(-2\pi+\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}\right)\sqrt{1+\left(-2\pi+\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}\right)^2}-\right.$$

$$\left.\text{ArcSinh}\left[\frac{2\pi\,\text{RIn1v}}{\text{hv}}\right]-\text{ArcSinh}\left[2\pi-\sqrt{\frac{4\,\text{L1v}\,\pi}{\text{hv}}+\frac{4\pi^2\,\text{RIn1v}^2}{\text{hv}^2}}\right]\right)$$

-----------------------------------------------------------

V. Export the data of the plots of the nanoscroll energy

as a function of the inner radius

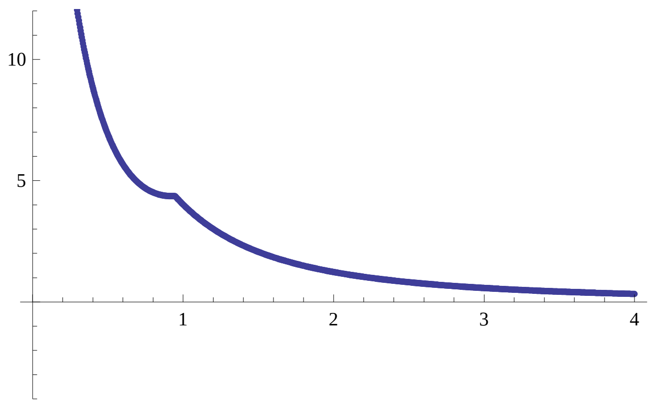
The parameters of the output file

The number of the output points = 1000

Export the plot data to the files:

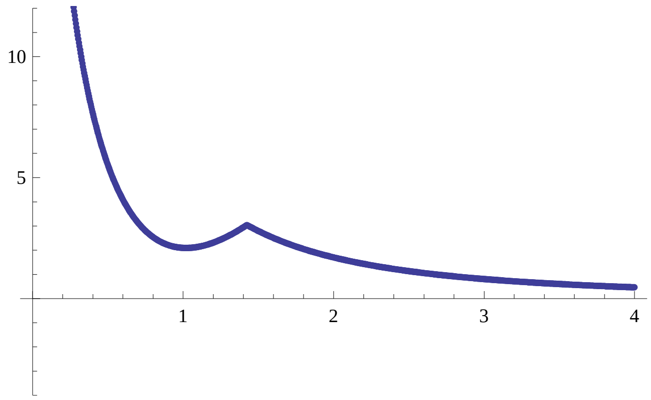
NanoscrollName=Nanoscroll1L7.nm

ScrollEnergyFileName=EvsRIn1Nanoscroll1L7.nm.dat



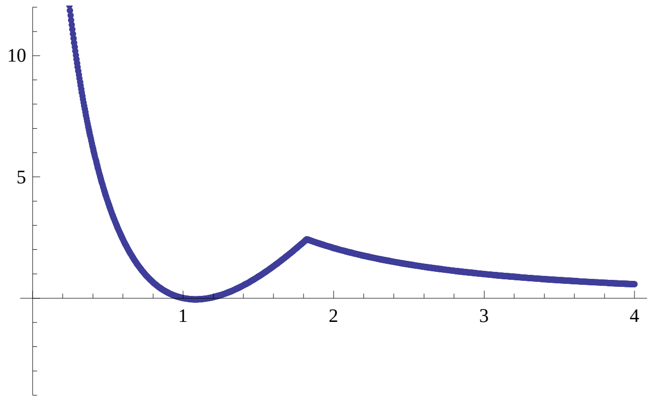
NanoscrollName=Nanoscroll1L10.nm

ScrollEnergyFileName=EvsRIn1Nanoscroll1L10.nm.dat



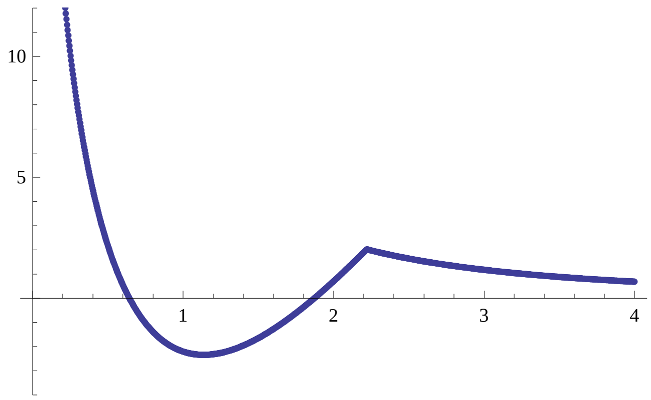

NanoscrollName=Nanoscroll1L12.5nm

ScrollEnergyFileName=EvsRIn1Nanoscroll1L12.5nm.dat

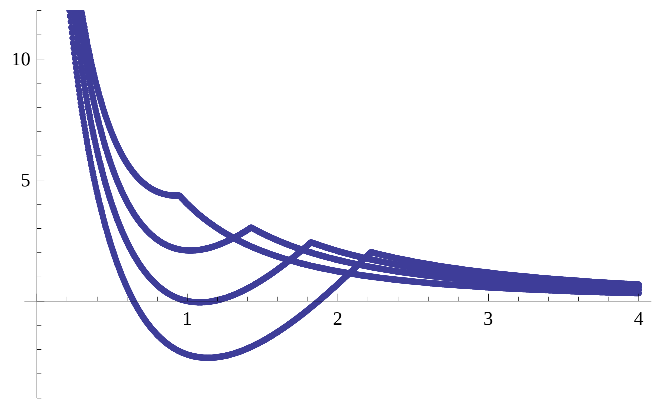



NanoscrollName=Nanoscroll1L15.nm

ScrollEnergyFileName=EvsRIn1Nanoscroll1L15.nm.dat




Plot ScrollEnergy[RIn1/nm]/(eV/atom) for L1={7., 10., 12.5, 15.}nm (NoL=1,Lw=1.nm)




```
Manipulating of the plot of the nanoscroll energy as the functoon
 of the nanoscroll inner radius:


(Manipulate[Plot[fScrollEnergy[...,L1nmm,RIn1,hp,...]]],
                          where L1nmm (is L1 in nanometers)
                          is the manipulated value
                         )
```

Out[556]=