In[ ]:=
```
<< Notation`
```

In[ ]:=
```
Notation[ x[[n_]] ⟺ x_n_ ]
```

In[ ]:=
```
(* All the numerical coefficients are defined at the beginning *)
a = 1.;
m = 1.;
ℏ = 0.1;
```

In[ ]:=
```
n = 20;
ξ = Range[-π/(2 a), π/(2 a), n];
(* The interval of n k-values sampled along the interval *)
```

In[ ]:=
```
2 t Cos[]
```

Cos : Cos called with 0 arguments; 1 argument is expected.

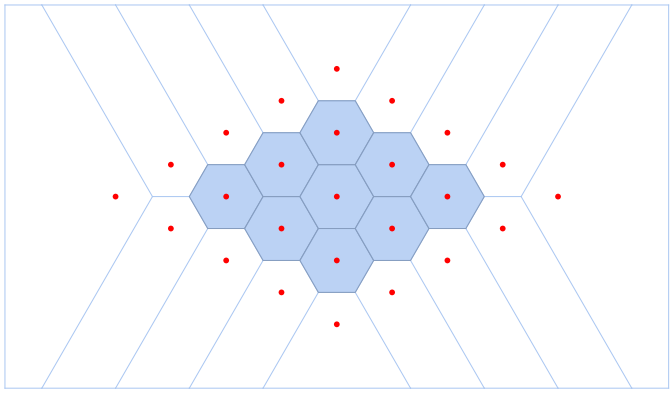
Out[ ]=
```
2 t Cos[]
```

In[ ]:=
```
Array[ℏ Norm[#]^2/(2 m) + # δ &, {
```

In[ ]:=
```
β = N@{a {√3/2, 1/2}, a {√3/2, -1/2}};
G = Tuples[Range[-2, 2], 2].β;
Length@G (* Number of sampled points *)
(* 1BZ represented for a honeycomb lattice *)
Show[VoronoiMesh[G, PlotTheme → "Lines",
  MeshCellStyle → {{2, "Interior"} → Directive[Opacity[0.8]]}],
 Graphics[{Red, Point[G]}], PlotRange → Automatic]
```

Out[ ]=
```
25
```

Out[ ]=



```
bas = N@{2 π/(a √3) {1, √3}, 2 π/(a √3) {1, -√3}};
path = N@With[{Γ = {0, 0}, K = {2/3, 1/3}.bas, M = {1/2, 1/2}.bas}, {M, Γ, K, M}];
(* The list of high-symmetry points *)
samplPts = Subdivide[#1, #2, n] &@@@ Partition[path, 2, 1] // Flatten[#, 1] & //
    DeleteAdjacentDuplicates;(* List of n points sampled along
  each line of the path going through the high-symmetry points,
it's literally the points generated by traversing the line,
although not in equal steps *)
Length@samplPts (* Total count of the sampling points *)
K = Tuples[Range[-1, 1], 2].bas;
Length@K (* Number of sampled points *)
Show[VoronoiMesh[K, PlotTheme → "Lines",
  MeshCellStyle → {{2, "Interior"} → Directive[Opacity[0.8]]}],
 Graphics[{{Red, Point[K]}, {Thick, Line[path]}}],
 PlotRange → Automatic, Axes → True]
Graphics[{Point[#] & /@ samplPts, Line[path]}, Axes → True]
```

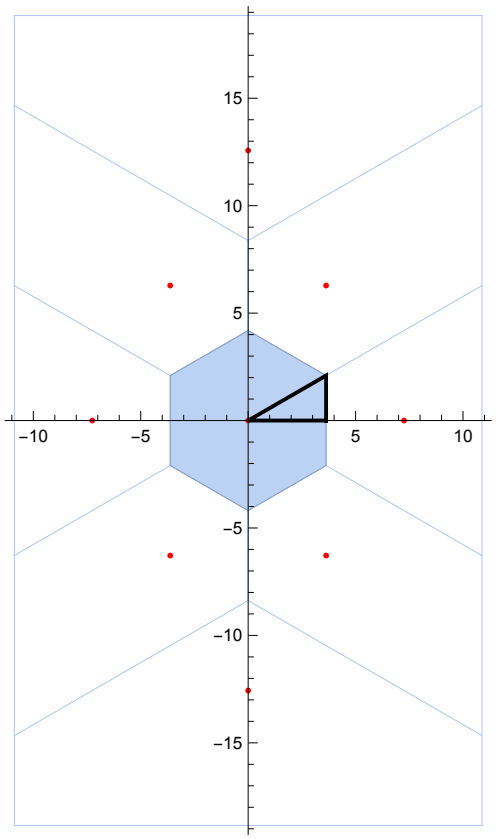
Out[ ]=

61

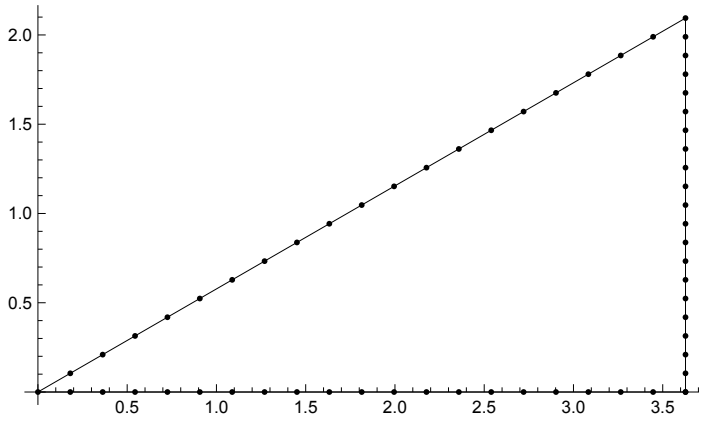
Out[ ]=

9

Out[ ]=

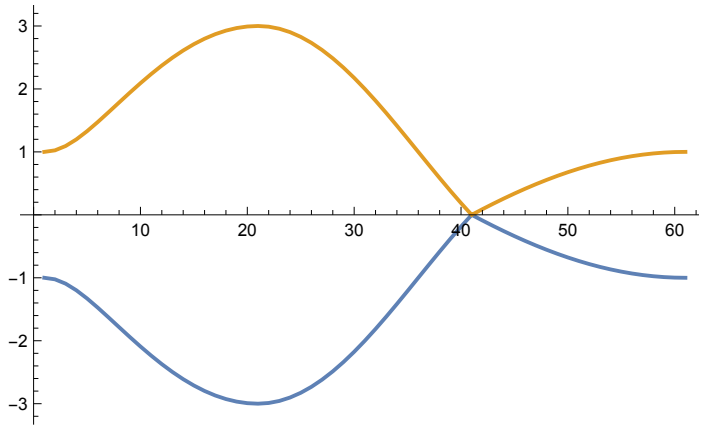


Out[ ]=




```
t = 1.;
fnband = t Sqrt[3 + 2 Cos[a #2] + 4 Cos[Sqrt[3]/2 a #1] Cos[a/2 #2]] & @@@ samplPts;
pm[a_] := {-a, a};
ListLinePlot[± fnband, PlotRange → Automatic]
```

Out[ ]=



In[ ]:=