```
(* Kronig-Penney model using transfer matrix *)

Clear[ell, q, p, epsilon, m]

In[1]:= ell := {{1, 1}, {I q, -I q}}   (* matrix L *)

In[2]:= MatrixForm[ell]
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 1 & 1 \\ i\,q & -i\,q \end{pmatrix}$$

```
In[3]:= v := {{1, 0}, {epsilon, 1}}  (* matrix V, epsilon as defined in Section 2.2 *)

In[4]:= MatrixForm[v]
```

Out[4]//MatrixForm=

$$\begin{pmatrix} 1 & 0 \\ \text{epsilon} & 1 \end{pmatrix}$$

```
p := {{Exp[I q], 0}, {0, Exp[-I q]}}  (* matrix P *)

In[6]:= MatrixForm[p]
```

Out[6]//MatrixForm=

$$\begin{pmatrix} e^{i\,q} & 0 \\ 0 & e^{-i\,q} \end{pmatrix}$$

```
In[7]:= t := p.Inverse[ell, Method -> "CofactorExpansion"].v.ell  (* transfer matrix T *)

MatrixForm[t]
```

$$\begin{pmatrix} e^{i\,q} - \frac{i\,e^{i\,q}\,\text{epsilon}}{2\,q} & -\frac{i\,e^{i\,q}\,\text{epsilon}}{2\,q} \\ \frac{i\,e^{-i\,q}\,\text{epsilon}}{2\,q} & e^{-i\,q} + \frac{i\,e^{-i\,q}\,\text{epsilon}}{2\,q} \end{pmatrix}$$

```
Det[t]
```

1

```
(* Eigenvectors of transfer matrix t *)

Eigenvectors[t]
```

$$\Big\{\Big\{-\frac{1}{2\,\text{epsilon}}\Big(\text{epsilon} + e^{2\,i\,q}\,\text{epsilon} - 2\,i\,q + 2\,i\,e^{2\,i\,q}\,q - i\sqrt{\left(-16\,e^{2\,i\,q}\,q^2 + \left(-i\,\text{epsilon} + i\,e^{2\,i\,q}\,\text{epsilon} - 2\,q - 2\,e^{2\,i\,q}\,q\right)^2\right)}\Big),\ 1\Big\},$$
$$\Big\{-\frac{1}{2\,\text{epsilon}}\Big(\text{epsilon} + e^{2\,i\,q}\,\text{epsilon} - 2\,i\,q + 2\,i\,e^{2\,i\,q}\,q + i\sqrt{\left(-16\,e^{2\,i\,q}\,q^2 + \left(-i\,\text{epsilon} + i\,e^{2\,i\,q}\,\text{epsilon} - 2\,q - 2\,e^{2\,i\,q}\,q\right)^2\right)}\Big),\ 1\Big\}\Big\}$$

```
(* pasted from above *)

In[8]:= va1[q_, epsilon_] := {-((epsilon + e^(2 i q) epsilon - 2 i q + 2 i e^(2 i q) q -
          i Sqrt[(-16 e^(2 i q) q^2 + (-i epsilon + i e^(2 i q) epsilon - 2 q - 2 e^(2 i q) q)^2)]) / (2 epsilon)), 1}

In[9]:= v1[q_, epsilon_] := va1[q, epsilon] / Norm[va1[q, epsilon]]   (* normalized *)
```

In[10]:=
```
va2[q_, epsilon_] := {- ((epsilon + ⅇ^(2 ⅈ q) epsilon - 2 ⅈ q + 2 ⅈ ⅇ^(2 ⅈ q) q +
      ⅈ √(-16 ⅇ^(2 ⅈ q) q^2 + (-ⅈ epsilon + ⅈ ⅇ^(2 ⅈ q) epsilon - 2 q - 2 ⅇ^(2 ⅈ q) q)^2)) / (2 epsilon)), 1}
```

In[11]:=
```
v2[q_, epsilon_] := va2[q, epsilon] / Norm[va2[q, epsilon]]
```

```
(* Simplified form of eigenvectors *)
(* Note: this notebook generally assumes that q = omega > 0 *)
```

In[12]:=
```
qq[q_, epsilon_] :=
 Cos[q] + (epsilon / (2 q)) Sin[q]   (* qq(q,epsilon) = cos(k) with k = wave vector *)
```

In[13]:=
```
ww[q_, epsilon_] := -Cos[q] + (2 q / epsilon) Sin[q]
```

In[14]:=
```
vv1[q_, epsilon_] :=
 {Exp[I q] (ww[q, epsilon] + (2 q / epsilon) Sqrt[1 - qq[q, epsilon] ^ 2]), 1}
```

In[15]:=
```
vv2[q_, epsilon_] :=
 {Exp[I q] (ww[q, epsilon] - (2 q / epsilon) Sqrt[1 - qq[q, epsilon] ^ 2]), 1}
```

```
(* Check that vv1 = va1, vv2 = va2 *)

va1[0.7, 0.1]

{12.5798 + 10.5958 ⅈ, 1}

vv1[0.7, 0.1]

{12.5798 + 10.5958 ⅈ, 1}

va2[0.7, 0.1]

{0.0465017 + 0.0391679 ⅈ, 1}

vv2[0.7, 0.1]

{0.0465017 + 0.0391679 ⅈ, 1}

(* End check *)
```

```
(* Modified, normalized eigenvectors to be used in the following *)
```

In[16]:=
```
v1norm[q_, epsilon_] := vv1[q, epsilon] / Norm[vv1[q, epsilon]]
```

In[17]:=
```
v2norm[q_, epsilon_] :=
 Exp[-I q] vv2[q, epsilon] / Norm[vv2[q, epsilon]]  (* note factor Exp[-I q] *)
```

```
(* v1norm[[1]]+v1norm[[2]] and v2norm[[1]]+v2norm[[2]] are complex conjugate,
   but only with factor Exp[-I q] in v2norm included. This is the modification. *)

v1norm[0.7, 0.1][[1]] + v1norm[0.7, 0.1][[2]]

0.82412 + 0.64303 ⅈ
```

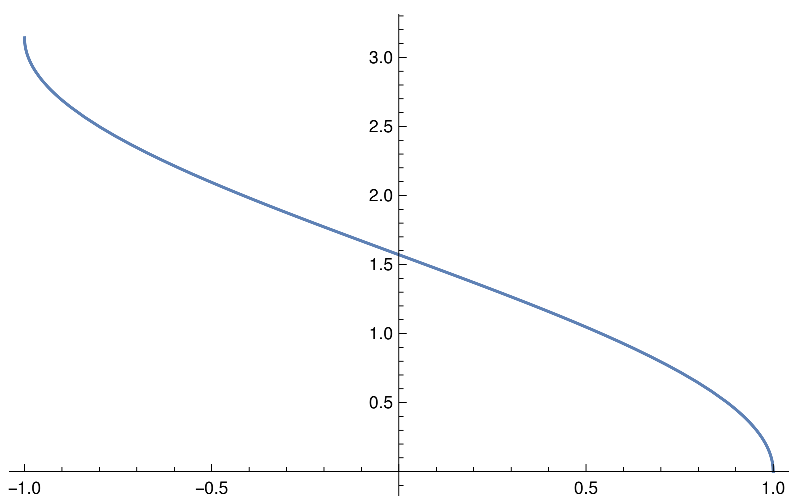
```
v2norm[0.7, 0.1][[1]] + v2norm[0.7, 0.1][[2]]
0.82412 - 0.64303 i

(* alternative form of v2norm, further simplified *)

In[18]:= vv2alt[q_, epsilon_] :=
          {ww[q, epsilon] - (2 q / epsilon) Sqrt[1 - qq[q, epsilon] ^2], Exp[-I q]}

In[19]:= v2normalt[q_, epsilon_] := vv2alt[q, epsilon] / Norm[vv2alt[q, epsilon]]

In[20]:= v2normalt[0.7, 0.1][[1]] + v2normalt[0.7, 0.1][[2]]
Out[20]= 0.82412 - 0.64303 i


(* Eigenvalues mu of transfer matrix t *)

In[21]:= Eigenvalues[t]
```

Out[21]=

$$\left\{\frac{1}{4\,q}e^{-i\,q}\left(i\,\text{epsilon}-i\,e^{2\,i\,q}\,\text{epsilon}+2\,q+2\,e^{2\,i\,q}\,q-\sqrt{\left(-16\,e^{2\,i\,q}\,q^2+\left(-i\,\text{epsilon}+i\,e^{2\,i\,q}\,\text{epsilon}-2\,q-2\,e^{2\,i\,q}\,q\right)^2\right)}\right),\right.$$
$$\left.\frac{1}{4\,q}e^{-i\,q}\left(i\,\text{epsilon}-i\,e^{2\,i\,q}\,\text{epsilon}+2\,q+2\,e^{2\,i\,q}\,q+\sqrt{\left(-16\,e^{2\,i\,q}\,q^2+\left(-i\,\text{epsilon}+i\,e^{2\,i\,q}\,\text{epsilon}-2\,q-2\,e^{2\,i\,q}\,q\right)^2\right)}\right)\right\}$$

```
(* => mu = exp(+/- i k), cos(k) = cos(q) + epsilon/(2q) sin(q) =: qq(q,epsilon) *)

In[22]:= Plot[ArcCos[x], {x, -1, 1}]
```

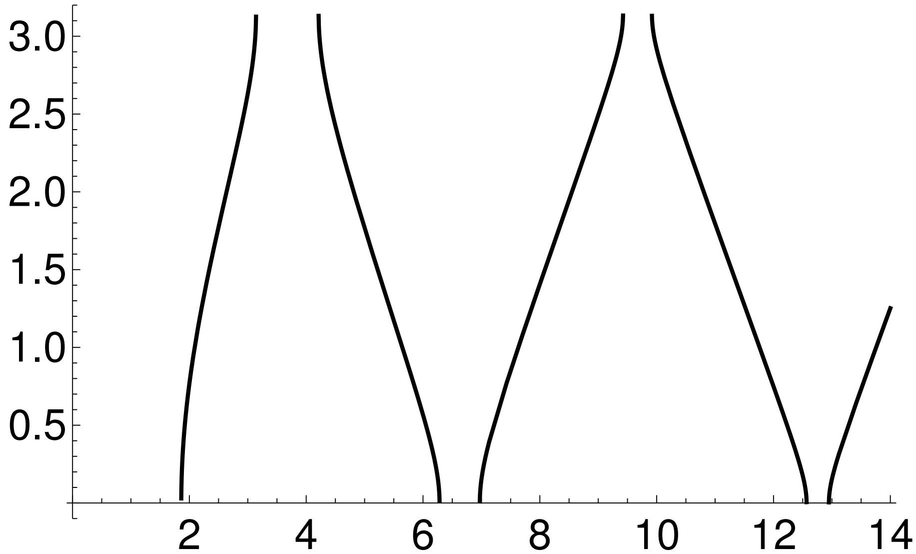
Out[22]=




```
In[23]:= kk[q_, epsilon_] := ArcCos[qq[q, epsilon]]
          (* k(q) wave vector, reduced range 0 < k < Pi *)

In[24]:= epsilon := 5

(* Plot band structure q(k) = omega(k) for epsilon = 5 *)
```

```
(* A) Reduced zone scheme with -Pi<k<Pi *)

In[86]:= komega = Plot[kk[q, epsilon], {q, 0, 14}, PlotRange → {{-0.1, 14.1}, {-0.1, 3.2}},
           PlotStyle → {Black, Thickness[0.005]}, AxesStyle → Directive[22]]
```

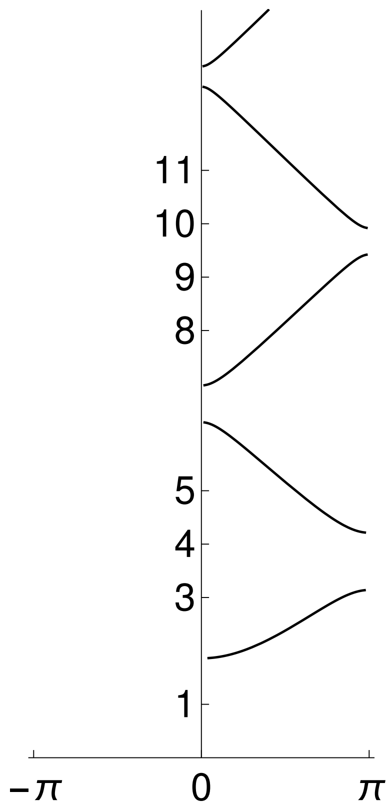
Out[86]=




```
In[87]:= Export["komega.png", komega, ImageResolution → 300]

Out[87]= komega.png

In[26]:= kkmod[q_, epsilon_] := If[Abs[qq[q, epsilon]] < 1, kk[q, epsilon], 100]
          (* k(q), allow only |qq|<1 *)

(* This is the form of k(q) = k(omega) > 0 used in program *)
(* corresponding to the reduced zone scheme. *)


(* Use parametric plot to plot inverse function q(k) = omega(k) *)
```

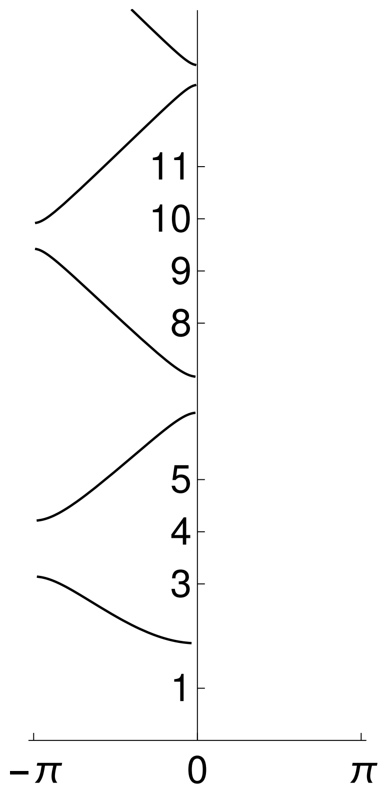
In[27]:=
```
pos = ParametricPlot[{kkmod[q, epsilon], q},
  {q, 0, 14}, PlotRange → {{-Pi - 0.1, Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.007]}}, AxesStyle → Directive[20],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {1, 3, 4, 5, 8, 9, 10, 11}}]
```

Out[27]=



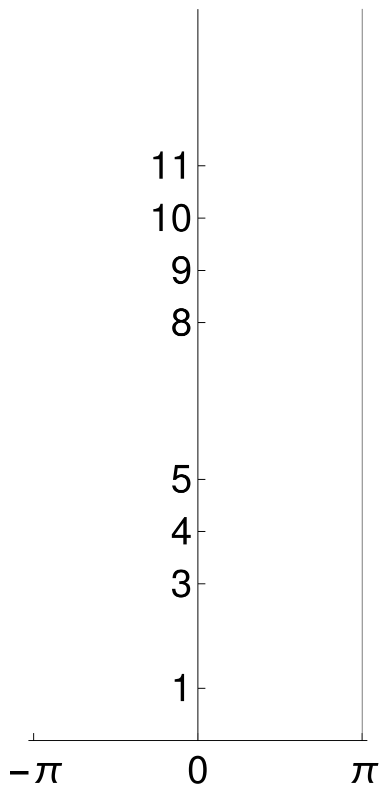
In[28]:=
```
neg = ParametricPlot[{-kkmod[q, epsilon], q},
  {q, 0, 14}, PlotRange → {{-Pi - 0.1, Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.007]}}, AxesStyle → Directive[20],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {1, 3, 4, 5, 8, 9, 10, 11}}]
```

Out[28]=



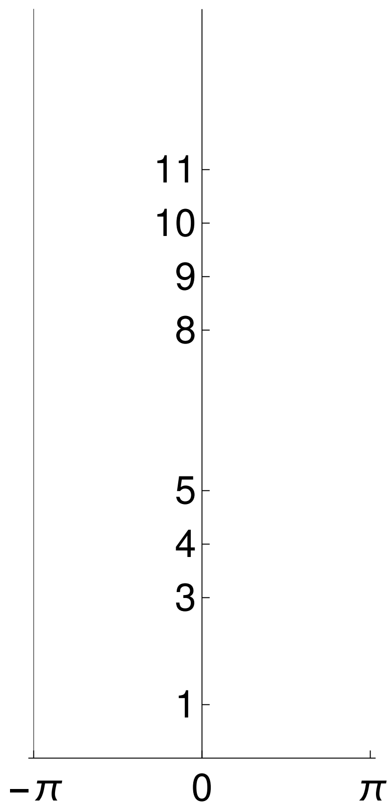
In[29]:=
```
pipos = ParametricPlot[{Pi, q}, {q, 0, 14}, PlotRange → {{-Pi - 0.1, Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.0015]}}, AxesStyle → Directive[20],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {1, 3, 4, 5, 8, 9, 10, 11}}]
```

Out[29]=



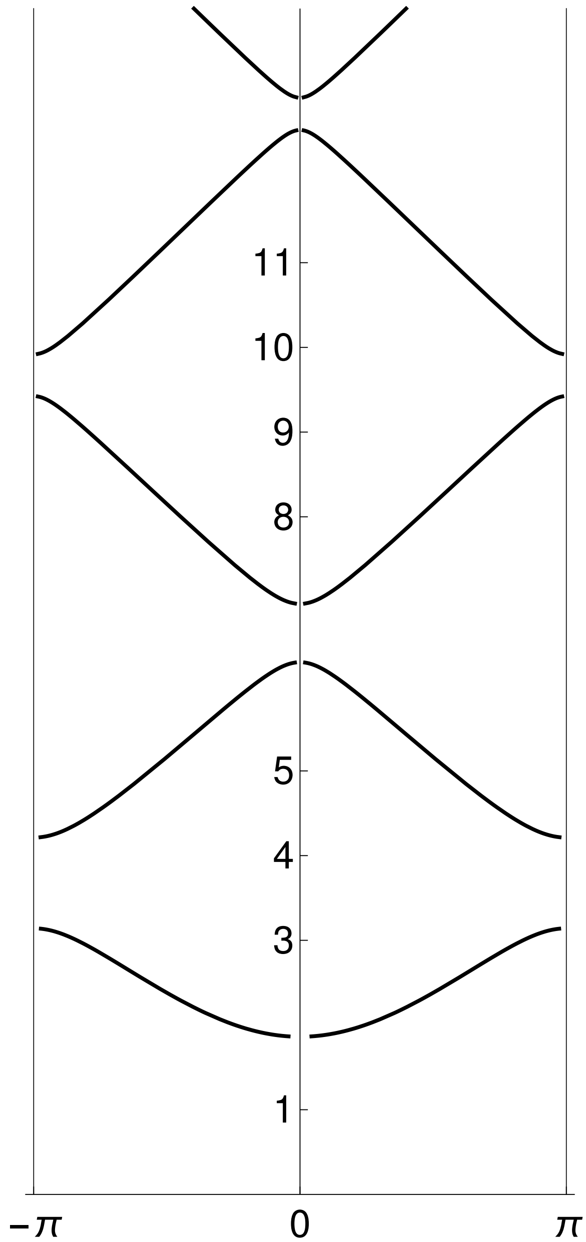
In[30]:=
```
pineg = ParametricPlot[{-Pi, q}, {q, 0, 14}, PlotRange → {{-Pi - 0.1, Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.0015]}}, AxesStyle → Directive[20],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {1, 3, 4, 5, 8, 9, 10, 11}}]
```

Out[30]=



```
bandreduced = Show[neg, pos, pipos, pineg]   (* dispersion relation q(k) = omega(k) *)
```

Out[31]=

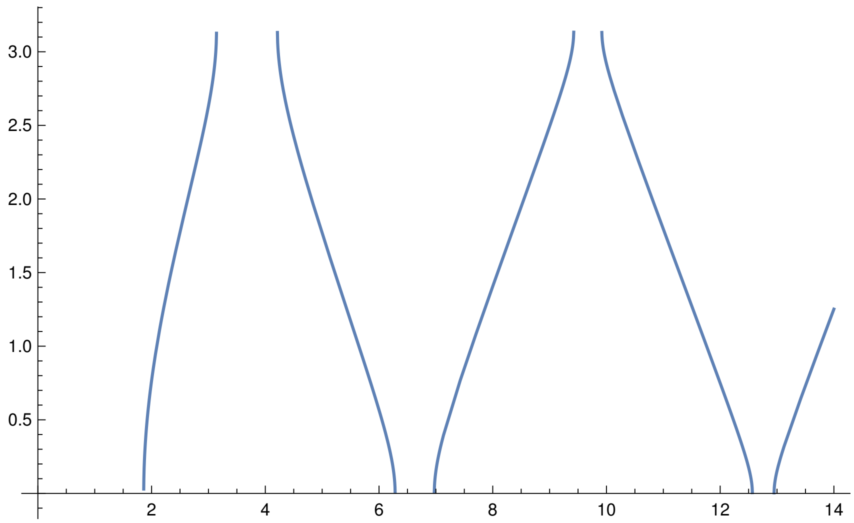



```
Export["bs_reduced.png", bandreduced, ImageResolution → 300]
```

bs_reduced.png

```
(* This is the dispersion relation q(k) = omega(k) used in the program *)
(* corresponding to reduced zone scheme. *)


(* Find gaps to 7 digits (for epsilon = 5) *)
```

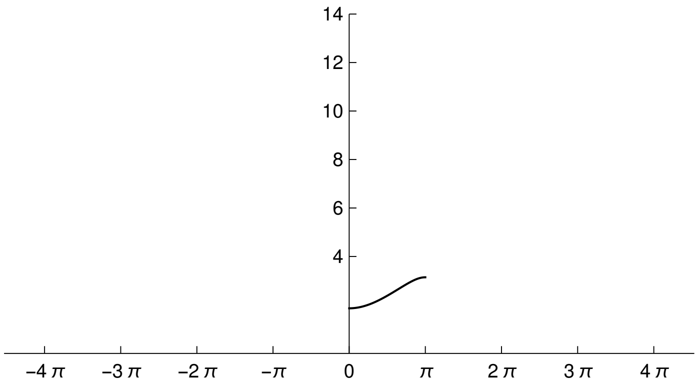
```
Plot[kk[q, epsilon], {q, 0, 14}]     (* k(q) wave vector, reduced range 0 < k < Pi *)
```



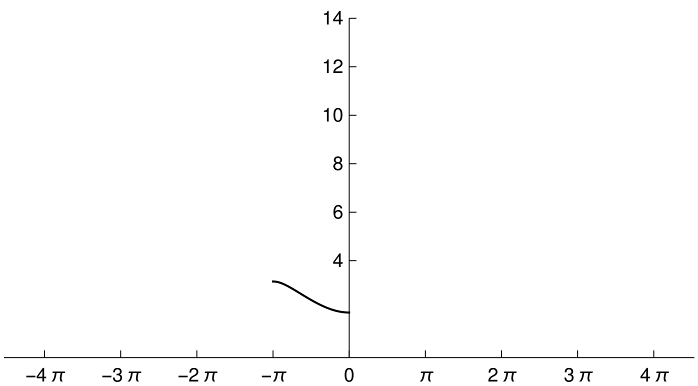

```
kk[1.861514, epsilon]      (* first allowed in 1. band *)
0.00149712

In[32]:= first1 := 1.861514          (* first allowed q=omega in 1. band *)

kk[Pi, epsilon]               (* last allowed in 1. band = Pi *)
π

In[33]:= last1 = Pi - 0.000001
Out[33]= 3.14159

kk[4.2127514, epsilon]    (* first allowed in 2. band *)
3.14132

In[34]:= first2 := 4.2127514

kk[2 Pi, epsilon]           (* last allowed in 2. band = 2 Pi *)
0

In[35]:= last2 = 2 Pi - 0.000001
Out[35]= 6.28318

kk[6.971795, epsilon]      (* first allowed in 3. band *)
0.000349891
```

```
In[36]:= first3 := 6.971795

kk[3 * Pi, epsilon]        (* last allowed in 3. band = 3 Pi *)
π

In[37]:= last3 = 3 Pi - 0.000001
Out[37]= 9.42478

kk[9.918596, epsilon]     (* first allowed in 4. band *)
3.14116

In[38]:= first4 := 9.918596

kk[4 Pi, epsilon]         (* last allowed in 4. band = 4 Pi *)
0

In[39]:= last4 = 4 Pi - 0.000001
Out[39]= 12.5664

kk[12.947842, epsilon]    (* first allowed in 5. band *)
0.000598452

In[40]:= first5 := 12.947842

In[41]:= last5 := 14

(* B) Extended zone scheme *)
(* Only for illustration,not used in program *)
```

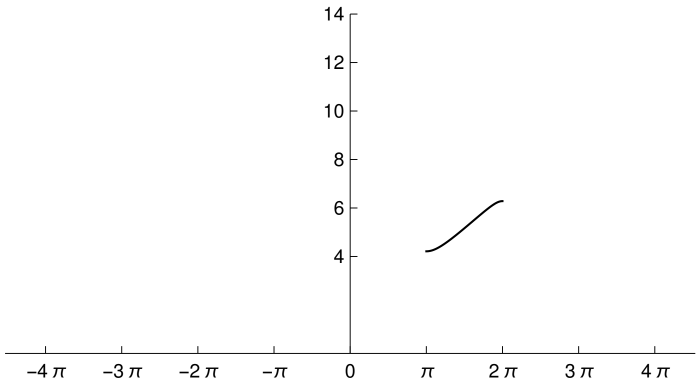
```
pos1 = ParametricPlot[{kkmod[q, epsilon], q},
  {q, first1, last1}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



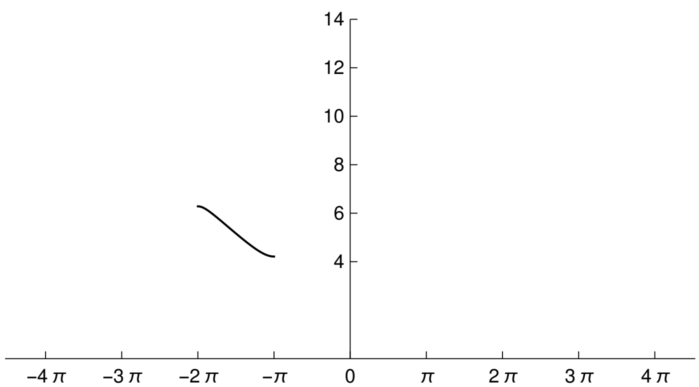

```
neg1 = ParametricPlot[{-kkmod[q, epsilon], q},
  {q, first1, last1}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



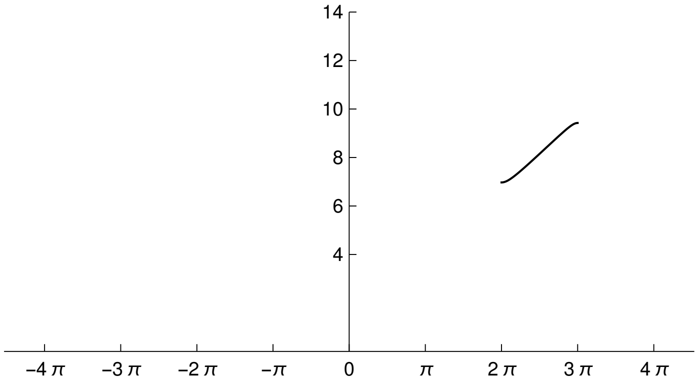
```
pos2 = ParametricPlot[{-kkmod[q, epsilon] + 2 Pi, q},
  {q, first2, last2}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



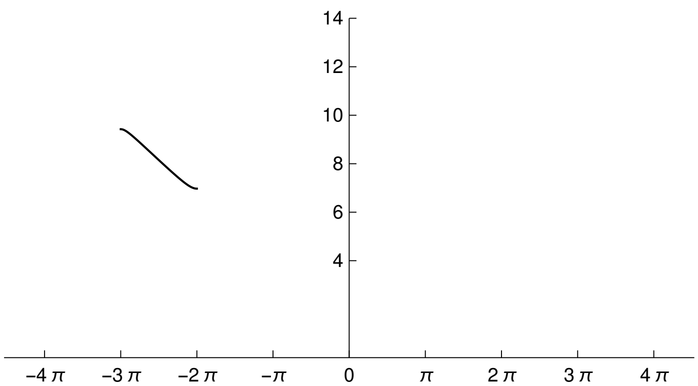

```
neg2 = ParametricPlot[{kkmod[q, epsilon] - 2 Pi, q},
  {q, first2, last2}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



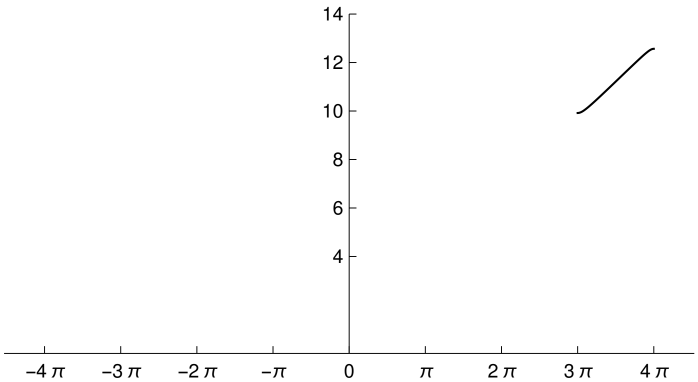
```
pos3 = ParametricPlot[{kkmod[q, epsilon] + 2 Pi, q},
  {q, first3, last3}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



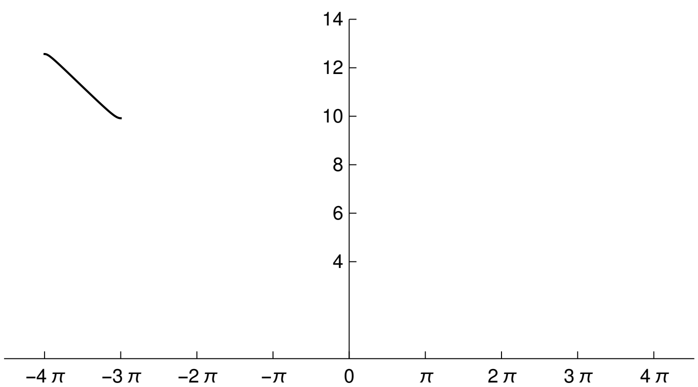

```
neg3 = ParametricPlot[{-kkmod[q, epsilon] - 2 Pi, q},
  {q, first3, last3}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



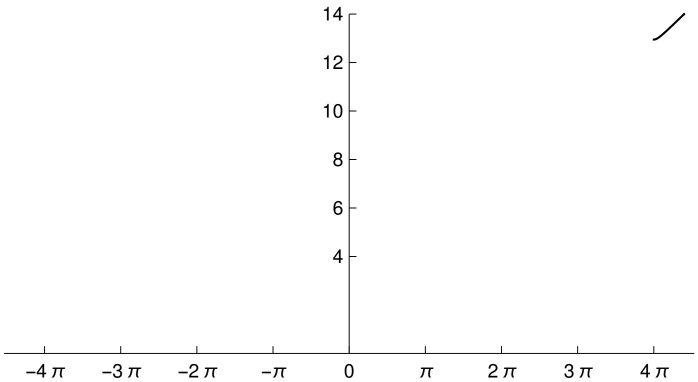
```
pos4 = ParametricPlot[{-kkmod[q, epsilon] + 4 Pi, q},
  {q, first4, last4}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



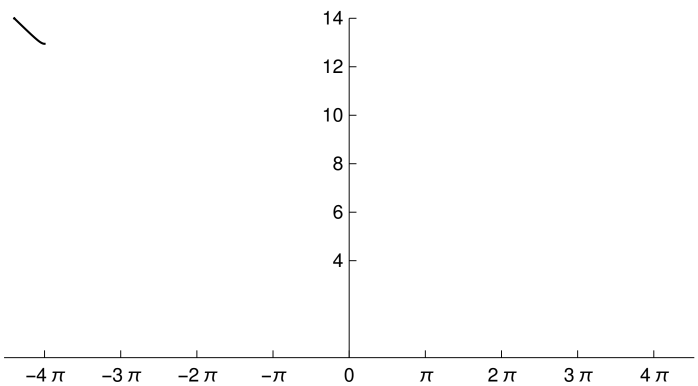

```
neg4 = ParametricPlot[{kkmod[q, epsilon] - 4 Pi, q},
  {q, first4, last4}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```



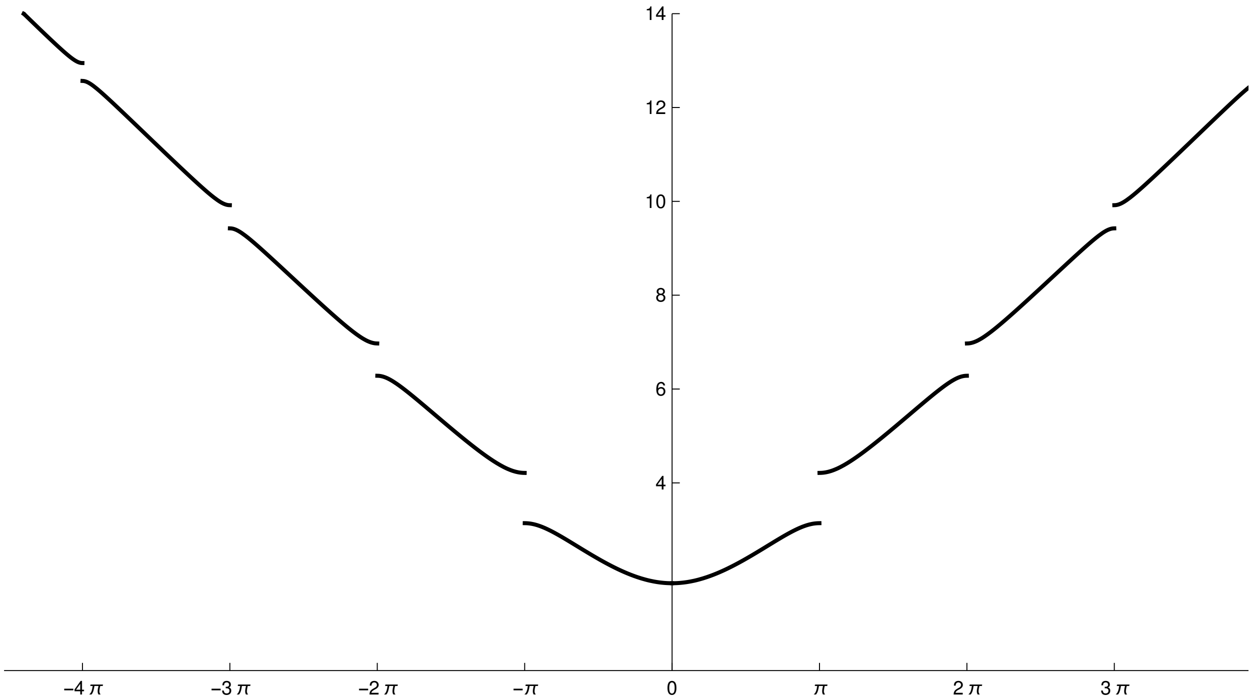
```
pos5 = ParametricPlot[{kkmod[q, epsilon] + 4 Pi, q},
  {q, first5, last5}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```




```
neg5 = ParametricPlot[{-kkmod[q, epsilon] - 4 Pi, q},
  {q, first5, last5}, PlotRange → {{-4.5 Pi - 0.1, 4.5 Pi + 0.1}, {0, 14}},
  PlotStyle → {{Black, Thickness[0.003]}}, AxesStyle → Directive[10],
  Ticks → {{-4 Pi, -3 Pi, -2 Pi, -Pi, 0, Pi, 2 Pi, 3 Pi, 4 Pi}, {4, 6, 8, 10, 12, 14}}]
```

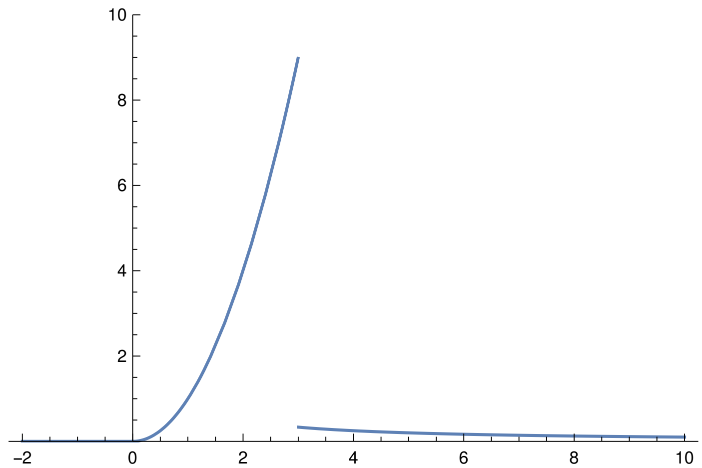


```
bandextended = Show[pos1, neg1, pos2, neg2, pos3, neg3, pos4, neg4, pos5, neg5]
```




```
(* Only for illustration, not used in program *)

Export["bs_extended.png", bandextended, ImageResolution → 300]

bs_extended.png

(* End Illustration *)

{first1, last1, first2, last2, first3, last3, first4, last4, first5, last5}

{1.86151, 3.14159, 4.21275, 6.28318, 6.9718, 9.42478, 9.9186, 12.5664, 12.9478, 14}


(* Now define function k(q) explicitly in extended zone scheme *)

(* Only for illustration, not used in program *)

test[x_] := Piecewise[{{x^2, x > 0 && x < 3}, {1 / x, x > 3}}, 0]
```

```
Plot[test[x], {x, -2, 10}, PlotRange → {0, 10}]
```

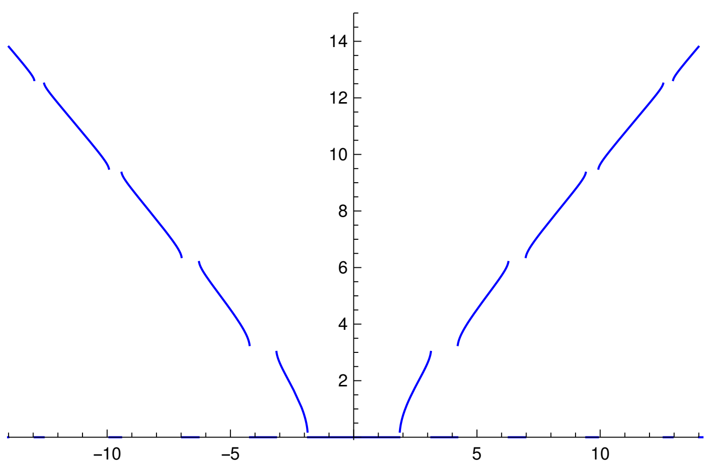



```
(* kkext[q] returns positive k[q]>0 for allowed q=
 omega and 0 for not allowed q=omega *)
```

```
kkext[q_, epsilon_] := Piecewise[{{kk[q, epsilon], Abs[q] > first1 && Abs[q] < last1},
    {-kk[q, epsilon] + 2 Pi, Abs[q] > first2 && Abs[q] < last2},
    {kk[q, epsilon] + 2 Pi, Abs[q] > first3 && Abs[q] < last3},
    {-kk[q, epsilon] + 4 Pi, Abs[q] > first4 && Abs[q] < last4},
    {kk[q, epsilon] + 4 Pi, Abs[q] > first5 && Abs[q] < last5}}, 0]
```

```
epsilon := 5
```

```
Plot[kkext[q, epsilon], {q, -15, 15},
 PlotRange -> {{-14, 14}, {-0.1, 15}}, PlotStyle → {Blue, Thickness[0.003]}]
```



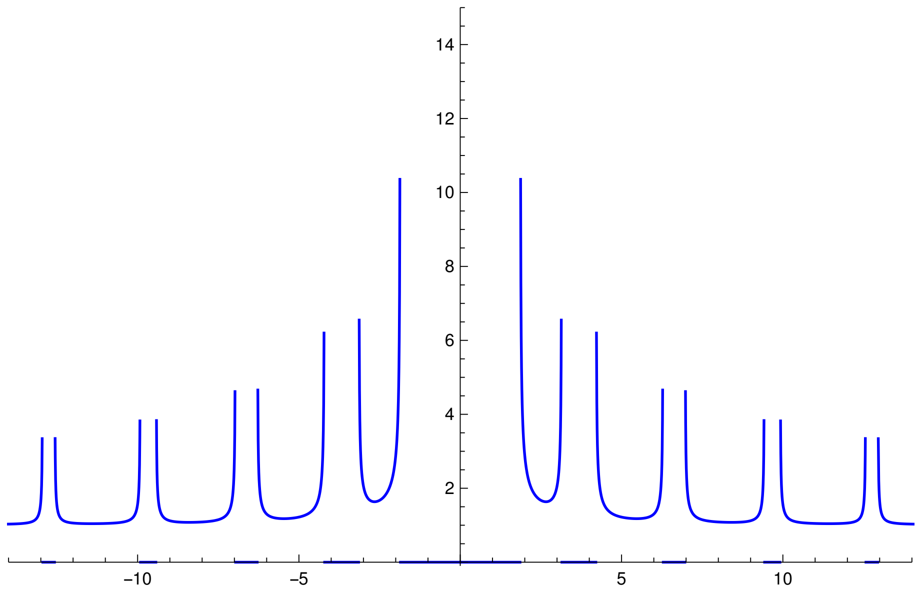
```
In[42]:= qq[q, eps]    (* qq(q,epsilon) = cos(k) with k = wave vector *)
```

Out[42]= $\text{Cos}[q] + \frac{\text{eps Sin}[q]}{2\,q}$

```
(* Derivative dk/dq *)

In[43]:= kkprime[q_, epsilon_] := Derivative[1, 0][kkmod][q, epsilon]

(* Analytical expression for Abs[dk/dq] *)

In[44]:= kkprimetest[q_, epsilon_] :=
  Abs[(1 + epsilon / (2 q^2)) Sin[q] - (epsilon / (2 q)) Cos[q]] / Sqrt[1 - qq[q, epsilon]^2]

In[45]:= epsilon := 5

In[46]:= kkprime[2.1, epsilon]    (* check in 1st band *)

Out[46]= 2.29166

In[47]:= kkprimetest[2.1, epsilon]

Out[47]= 2.29166

In[48]:= kkprime[4.7, epsilon]    (* check in 2nd band *)

Out[48]= -1.31896

In[49]:= kkprimetest[4.7, epsilon]

Out[49]= 1.31896

In[50]:= kkprime[7.5, epsilon]    (* check in 3rd band *)

Out[50]= 1.14931

In[51]:= kkprimetest[7.5, epsilon]

Out[51]= 1.14931

In[52]:= kkprime[11.2, epsilon]    (* check in 4th band *)

Out[52]= -1.04413

In[53]:= kkprimetest[11.2, epsilon]

Out[53]= 1.04413
```

In[54]:=
```
Plot[Abs[kkprime[q, epsilon]], {q, -15, 15},
 PlotRange -> {{-14, 14}, {-0.1, 15}}, PlotStyle → {Blue, Thickness[0.003]}]
```

Out[54]=

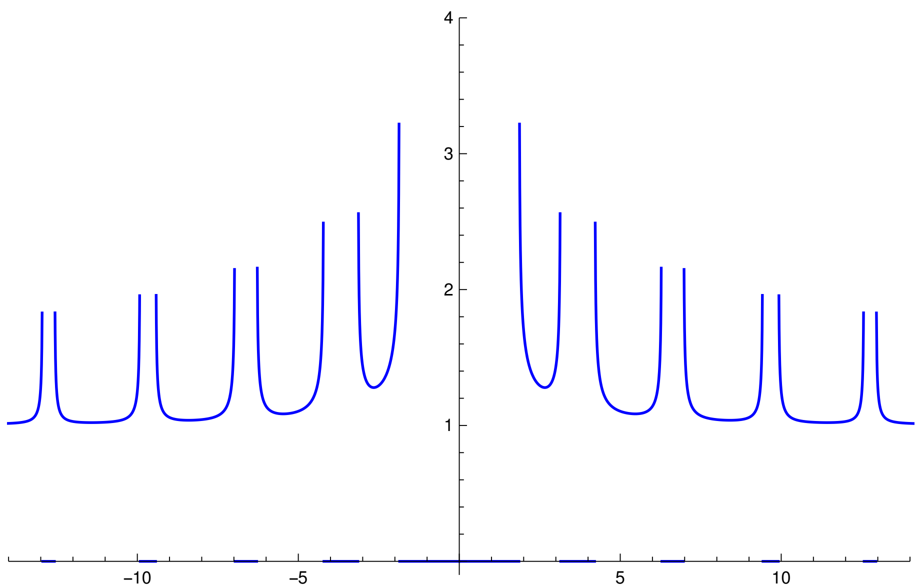


```
(* Only allowed q=omega should be used *)
```

In[55]:=
```
domega[q_, epsilon_] := Sqrt[Abs[kkprime[q, epsilon]]]   (* function domega(q) *)
```

In[56]:=
```
Plot[domega[q, epsilon], {q, -15, 15},
 PlotRange -> {{-14, 14}, {-0.1, 4}}, PlotStyle → {Blue, Thickness[0.003]}]
```

Out[56]=



```
(* Bloch functions Phi(x) with additional normalization *)

(* allow only |qq|<1 where qq = Cos[q]+(epsilon/(2q))Sin[q] *)
```

```
(* in what follows: index 1 = +k, index 2 = -k *)

In[57]:= exp1[q_, epsilon_, n_] :=
          If[Abs[qq[q, epsilon]] < 1, Exp[I kk[q, epsilon] n], 0]   (* k1 > 0 *)

In[58]:= exp2[q_, epsilon_, n_] :=
          If[Abs[qq[q, epsilon]] < 1, Exp[-I kk[q, epsilon] n], 0]  (* k2 = -k1 < 0 *)

         (* phi1raw, phi2raw not correctly normalized,
         missing normalization factors norm1,  norm2 below *)

In[59]:= phi1raw[x_, n_, q_, epsilon_] := exp1[q, epsilon, n]
           (v1norm[q, epsilon][[1]] Exp[I q (x - n)] + v1norm[q, epsilon][[2]] Exp[-I q (x - n)]) /
            (v1norm[q, epsilon][[1]] + v1norm[q, epsilon][[2]])

In[60]:= phi2raw[x_, n_, q_, epsilon_] := exp2[q, epsilon, n]
           (v2norm[q, epsilon][[1]] Exp[I q (x - n)] + v2norm[q, epsilon][[2]] Exp[-I q (x - n)]) /
            (v2norm[q, epsilon][[1]] + v2norm[q, epsilon][[2]])



In[61]:= epsilon := 5

         (* functions u(x) for 0<x<1, then periodically extended *)

In[62]:= u1raw[x_, q_, epsilon_] := (v1norm[q, epsilon][[1]] Exp[I (q - kk[q, epsilon]) (x - 1)] +
              v1norm[q, epsilon][[2]] Exp[-I (q + kk[q, epsilon]) (x - 1)]) /
           (v1norm[q, epsilon][[1]] + v1norm[q, epsilon][[2]])

In[63]:= norm1[q_, epsilon_]  :=
          Sqrt[NIntegrate[u1raw[x, q, epsilon] Conjugate[u1raw[x, q, epsilon]], {x, 0, 1}]]

         (* u1 correctly normalized used in program *)

In[64]:= u1[x_, q_, epsilon_]  := u1raw[x, q, epsilon] / norm1[q, epsilon]

         NIntegrate[u1[x, 2.1, epsilon] Conjugate[u1[x, 2.1, epsilon]], {x, 0, 1}]

Out[65]= 1.

In[66]:= u2raw[x_, q_, epsilon_] := (v2norm[q, epsilon][[1]] Exp[I (q + kk[q, epsilon]) (x - 1)] +
              v2norm[q, epsilon][[2]] Exp[-I (q - kk[q, epsilon]) (x - 1)]) /
           (v2norm[q, epsilon][[1]] + v2norm[q, epsilon][[2]])

In[67]:= norm2[q_, epsilon_]  :=
          Sqrt[NIntegrate[u2raw[x, q, epsilon] Conjugate[u2raw[x, q, epsilon]], {x, 0, 1}]]

         (* u2 correctly normalized used in program *)

In[68]:= u2[x_, q_, epsilon_]  := u2raw[x, q, epsilon] / norm2[q, epsilon]
```

```
NIntegrate[u2[x, 4.7, epsilon] Conjugate[u2[x, 4.7, epsilon]], {x, 0, 1}]
```
Out[69]= 1.

```
(* Show u_{-k}(x) = u_k^*(x) for all x; u1 is for k, u2 is for -k  *)
u1[0.7, 2.1, epsilon]
```
1.07325 - 0.0663097 i

```
u2[0.7, 2.1, epsilon]
```
1.07325 + 0.0663097 i

```
u1[3.25, 4.71, epsilon]
```
-0.72853 + 0.160454 i

```
u2[3.25, 4.71, epsilon]
```
-0.72853 - 0.160454 i

```
In[70]:= (* Show u(0) = u(1) real and equal for k, -k *)
In[71]:= eps := 5
In[72]:= q := 2.1
In[73]:= u1[0, q, eps]
```
Out[73]= $0.664728 - 9.5709 \times 10^{-17}\ i$

```
In[74]:= u1[1, q, eps]
```
Out[74]= $0.664728 + 1.15312 \times 10^{-18}\ i$

```
In[75]:= u2[0, q, eps]
```
Out[75]= $0.664728 + 9.5709 \times 10^{-17}\ i$

```
In[76]:= u2[1, q, eps]
```
Out[76]= 0.664728 + 0. i

```
q := 4.71        (* 2nd branch *)
u1[0, q, eps]
```
$0.948549 - 3.1593 \times 10^{-16}\ i$

```
u1[1, q, eps]
```
$0.948549 - 5.26551 \times 10^{-17}\ i$

```
u2[0, q, eps]
```
$0.948549 + 3.1593 \times 10^{-16}\ i$

```
u2[1, q, eps]
0.948549 + 0. i

(* Show u1=u2=1 for epsilon=0 and first branch *)

q = 2.1   (* 1st branch *)
2.1

u1[2.1, q, 0.0000000001]
1. + 3.22605 × 10^-11 i

u2[0.9, q, 0.0000000001]
1. + 2.38087 × 10^-12 i

q = 4.71   (* 2nd branch *)
4.71

u1[2.1, q, 0.0000000001]
0.809017 - 0.587785 i

u2[0.9, q, 0.0000000001]
0.809017 - 0.587785 i


(* functions c(q), d(q) *)

In[77]:= u1prime[x_, q_, epsilon_] :=
          Derivative[1, 0, 0][u1raw][x, q, epsilon] / norm1[q, epsilon]  (* k>0 *)

In[78]:= u2prime[x_, q_, epsilon_] :=
          Derivative[1, 0, 0][u2raw][x, q, epsilon] / norm2[q, epsilon]  (* k<0 *)

In[79]:= (* Show epsilon = [u'(0) - u'(1)]/u(0)  for k, -k *)

In[80]:= eps := 5

q := 2.1

(u1prime[0, q, eps] - u1prime[1, q, eps]) / u1[0, q, eps]
5. + 5.1834 × 10^-17 i

(u2prime[0, q, eps] - u2prime[1, q, eps]) / u1[0, q, eps]
5. + 2.3901 × 10^-15 i

q := 4.71

(u1prime[0, q, eps] - u1prime[1, q, eps]) / u1[0, q, eps]
5. - 2.07375 × 10^-16 i
```

```
(u2prime[0, q, eps] - u2prime[1, q, eps]) / u1[0, q, eps]
```

$5. + 2.60169 \times 10^{-15}\,i$

```
q := 7.1
(u1prime[0, q, eps] - u1prime[1, q, eps]) / u1[0, q, eps]
```

$5. + 2.58146 \times 10^{-16}\,i$

```
(u2prime[0, q, eps] - u2prime[1, q, eps]) / u1[0, q, eps]
```

$5. + 1.76432 \times 10^{-15}\,i$

```
(* end check *)

eps := 5
q := 2.1
u1prime[0, q, eps]
```

$1.66182 + 0.700078\,i$

```
u1prime[1, q, eps]
```

$-1.66182 + 0.700078\,i$

```
u2prime[0, q, eps]
```

$1.66182 - 0.700078\,i$

```
u2prime[1, q, eps]
```

$-1.66182 - 0.700078\,i$

```
q := 4.71
u1prime[0, q, eps]
```

$2.37137 - 5.80308\,i$

```
u1prime[1, q, eps]
```

$-2.37137 - 5.80308\,i$

```
u2prime[0, q, eps]
```

$2.37137 + 5.80308\,i$

```
u2prime[1, q, eps]
```

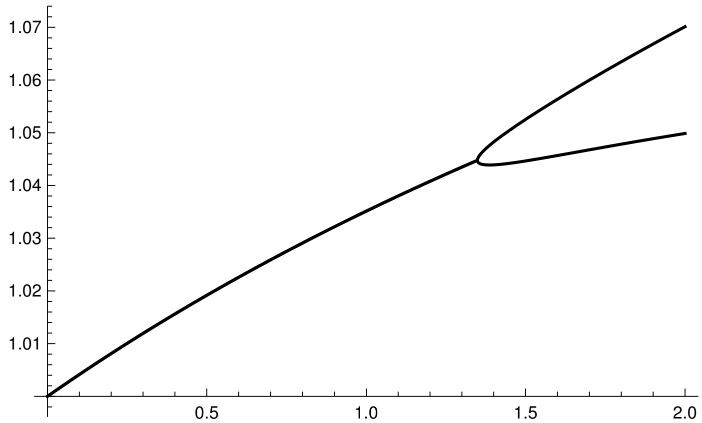
$-2.37137 + 5.80308\,i$

```
In[81]:= c[q_, epsilon_] := u1[0, q, epsilon]  (* cal C *)
(* c real and equal for (1), (2) *)
eps := 5
q := 2.1
c[q, eps]
0.664728 - 9.5709 × 10^-17 i
u2[0, q, eps]
0.664728 + 9.5709 × 10^-17 i
u1[1, q, epsilon]
0.664728 + 1.15312 × 10^-18 i
u2[1, q, epsilon]
0.664728 + 0. i
c[2, 3]
0.762883 + 3.46945 × 10^-17 i
u2[0, 2, 3]
0.762883 - 4.16334 × 10^-17 i
d[q_, epsilon_] := u1prime[0, q, epsilon]   (* cal D *)
(* d complex and conjugate for (1), (2) *)
eps := 5
q := 2.1
d[q, eps]
1.66182 + 0.700078 i
u2prime[0, q, eps]
1.66182 - 0.700078 i
d[4.71, eps]
2.37137 - 5.80308 i
u2prime[0, 4.71, eps]
2.37137 + 5.80308 i
```

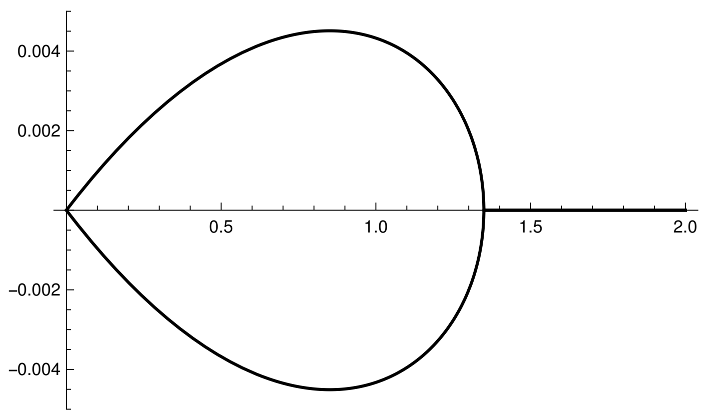
```
Abs[qq[1.1, 1.348]]  (* epsilon=1.348 is critical value for q=1.1 *)
```

0.999663

```
Plot[{Re[u1[0.4, 1.1, eps]], Re[u2[0.4, 1.1, eps]]}, {eps, 0, 2}, PlotStyle → Black]
```

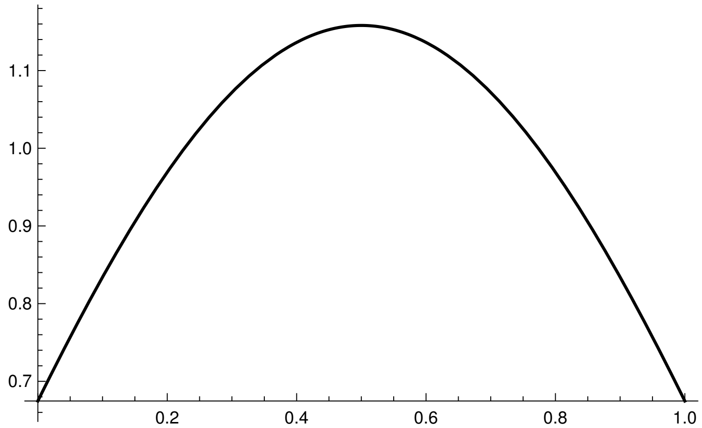



```
Plot[{Im[u1[0.4, 1.1, eps]], Im[u2[0.4, 1.1, eps]]}, {eps, 0, 2}, PlotStyle → Black]
```




```
(* Plot real part of u(x) for epsilon=5, q=omega=2 *)

eps := 5

q := 2
```

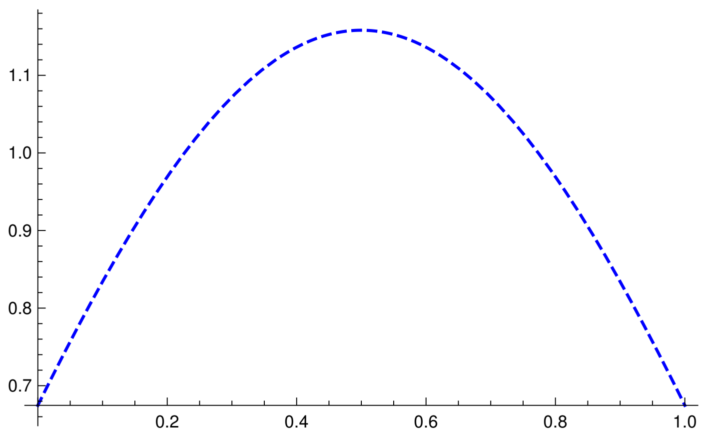
```
pu1 = Plot[Re[u1[x, q, eps]], {x, 0, 1}, PlotStyle → Black]
```



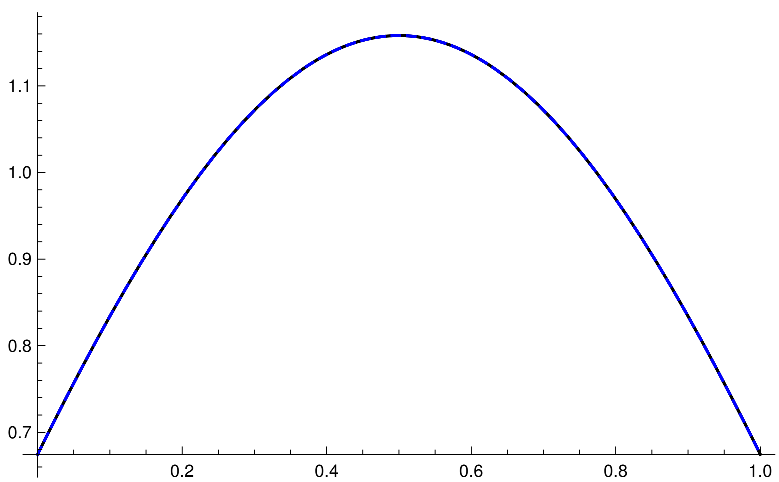

```
Export["u_real_eps5_omega2.png", pu1, ImageResolution → 300]
```

```
u_real_eps5_omega2.png
```

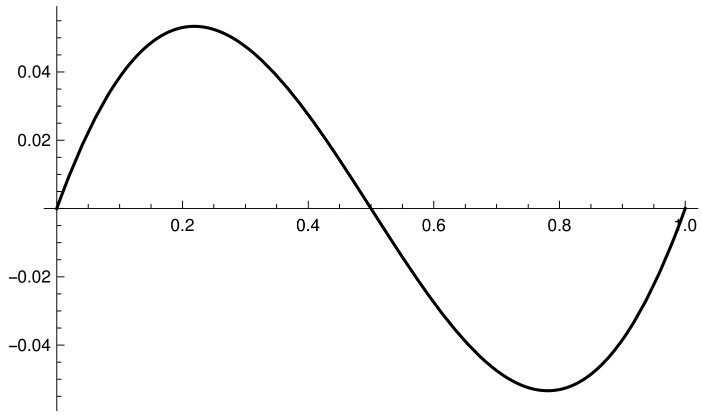
```
pu2 = Plot[Re[u2[x, q, eps]], {x, 0, 1}, PlotStyle → {Blue, Dashed}]
```



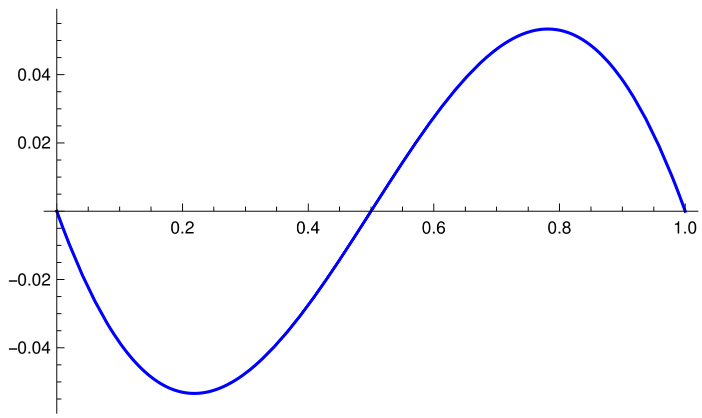

```
Show[pu1, pu2]
```



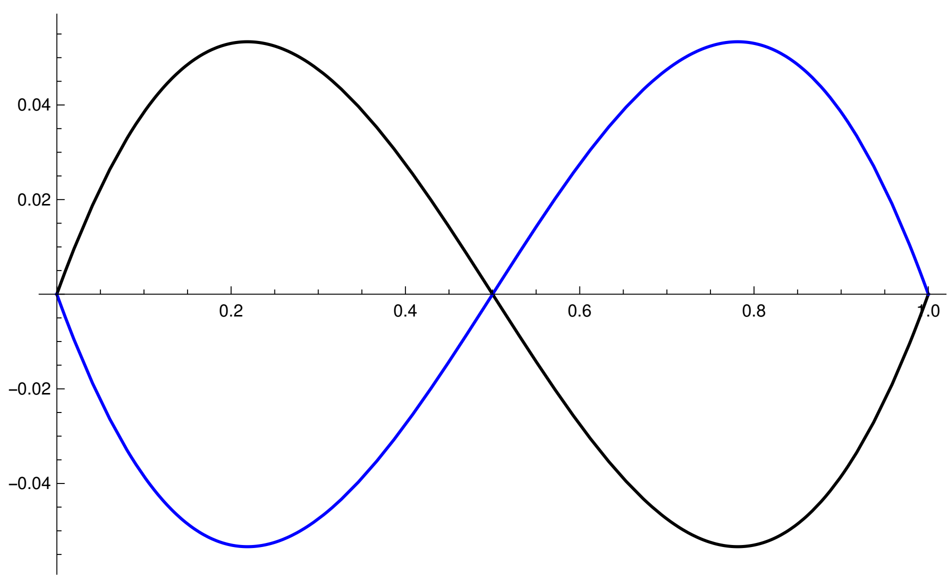
```
(* plot imaginary part of u(x) *)
pu3 = Plot[Im[u1[x, q, eps]], {x, 0, 1}, PlotStyle → Black]
```




```
pu4 = Plot[Im[u2[x, q, eps]], {x, 0, 1}, PlotStyle → {Blue}]
```



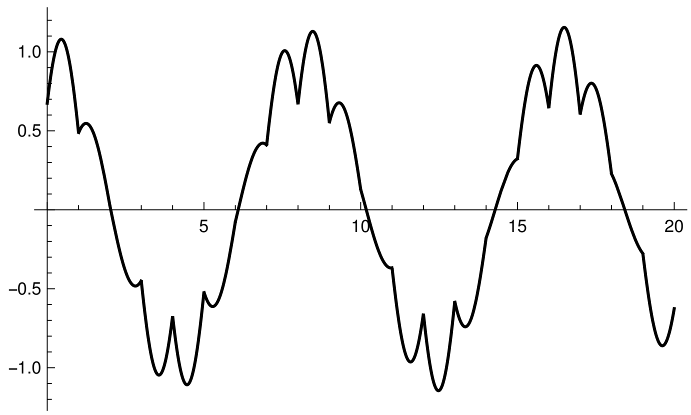

```
uim = Show[pu3, pu4]
```



```
Export["u_imaginary_eps5_omega2.png", uim, ImageResolution → 300]
```

```
u_imaginary_eps5_omega2.png
```

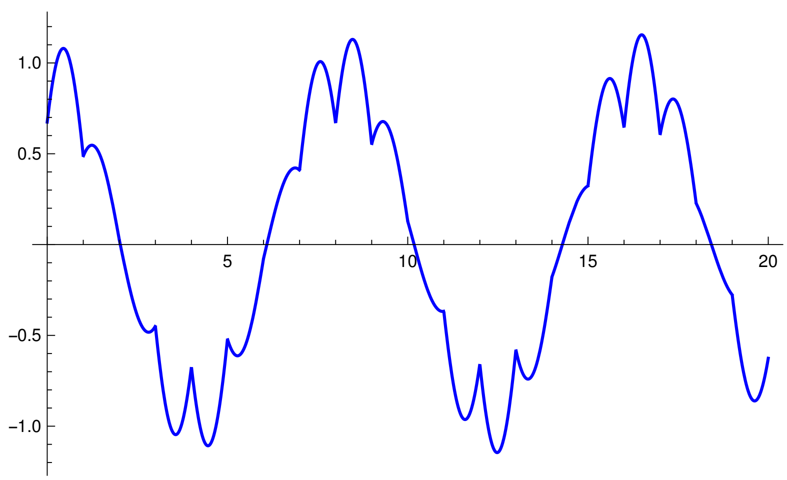
```
(* Plot real and imaginary parts of Bloch functions Phi(x) *)
(* Show Phi_2 = complex conjugate Phi_1 *)
eps := 5
q := 2
p1 = Plot[Re[phi1raw[x, Ceiling[x], q, eps]] / norm1[q, epsilon] ,
  {x, 0, 20}, PlotStyle → Black]
```



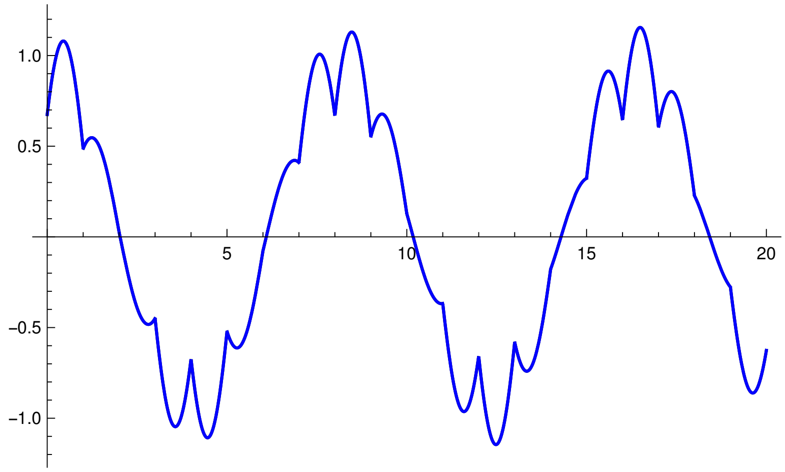

```
Export["Bloch_real_eps5_omega2.png", p1, ImageResolution → 300]
```

```
Bloch_real_eps5_omega2.png
```

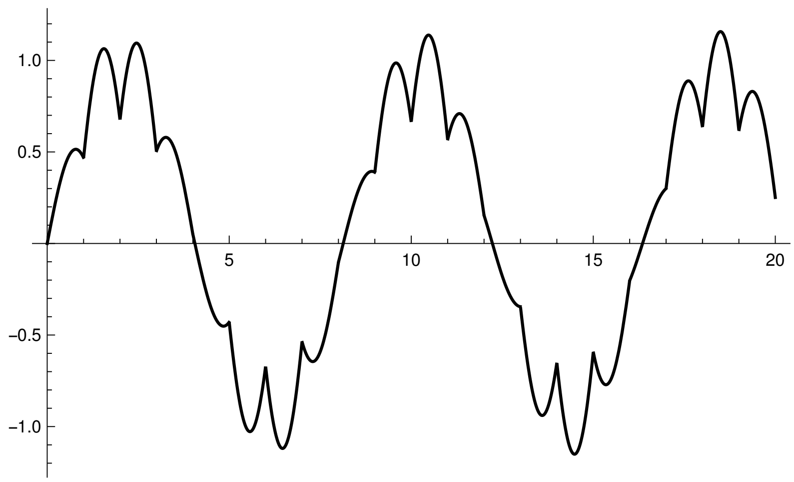
```
p2 = Plot[Re[phi2raw[x, Ceiling[x], q, eps]] / norm2[q, epsilon],
  {x, 0, 20}, PlotStyle → Blue]    (* same as Re(phi1) *)
```



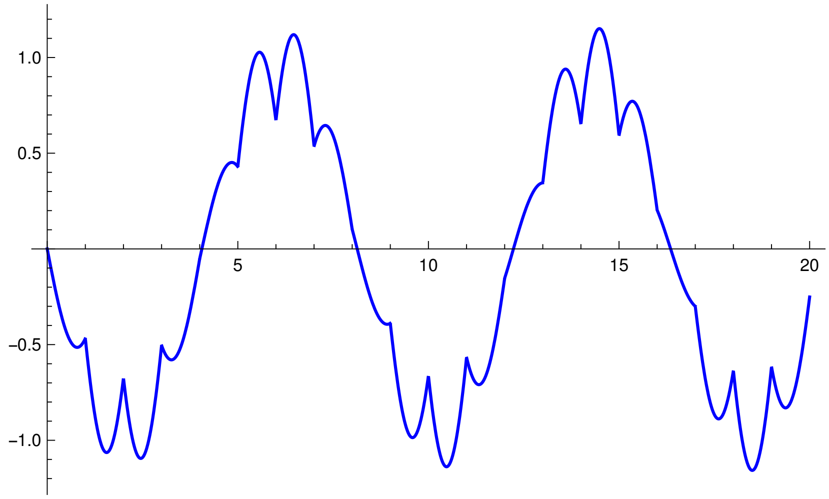
```
Show[p1, p2]
```



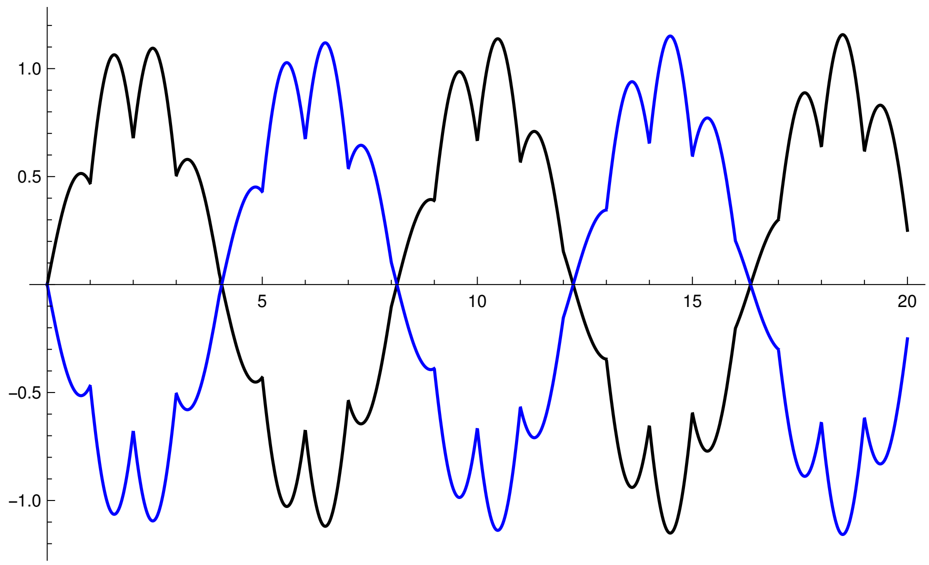

```
p3 = Plot[Im[phi1raw[x, Ceiling[x], q, eps]] / norm1[q, epsilon],
  {x, 0, 20}, PlotStyle → Black]
```



```
p4 = Plot[Im[phi2raw[x, Ceiling[x], q, eps]] / norm2[q, epsilon],
  {x, 0, 20}, PlotStyle → Blue]   (* -Im(phi1raw) *)
```




```
blochim = Show[p3, p4]
```




```
Export["Bloch_imaginary_eps5_omega2.png", blochim, ImageResolution → 300]
```

```
Bloch_imaginary_eps5_omega2.png
```

```
(* Show that phi1raw(x=n,n,q,epsilon) = Exp(i k n) *)

eps := 5

q := 2

phi1raw[7, 7, q, eps] // N
```

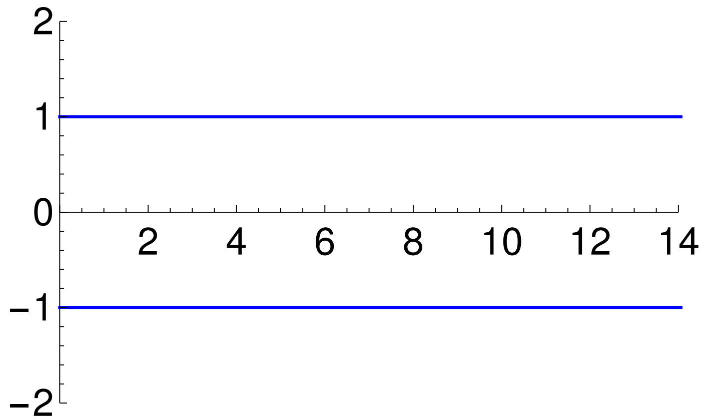
```
0.6066 - 0.795007 i
```

```
Exp[I kk[q, eps] 7] // N
0.6066 - 0.795007 i

phi1raw[8, 8, q, eps] // N
0.988362 - 0.152117 i

Exp[I kk[q, eps] 8] // N
0.988362 - 0.152117 i


(* Show that phi2raw(x=n,n,q,epsilon) = Exp(-i k n) *)

eps := 5

q := 2

phi2raw[7, 7, q, eps] // N
0.6066 + 0.795007 i

Exp[-I kk[q, eps] 7] // N
0.6066 + 0.795007 i

phi2raw[8, 8, q, eps] // N
0.988362 + 0.152117 i

Exp[-I kk[q, eps] 8] // N
0.988362 + 0.152117 i


(* Illustration: Function cos(k) *)

Clear[q, epsilon]

f[q_] := Cos[q] + epsilon / (2 q) Sin[q]     (* q = omega *)

f[q]
```

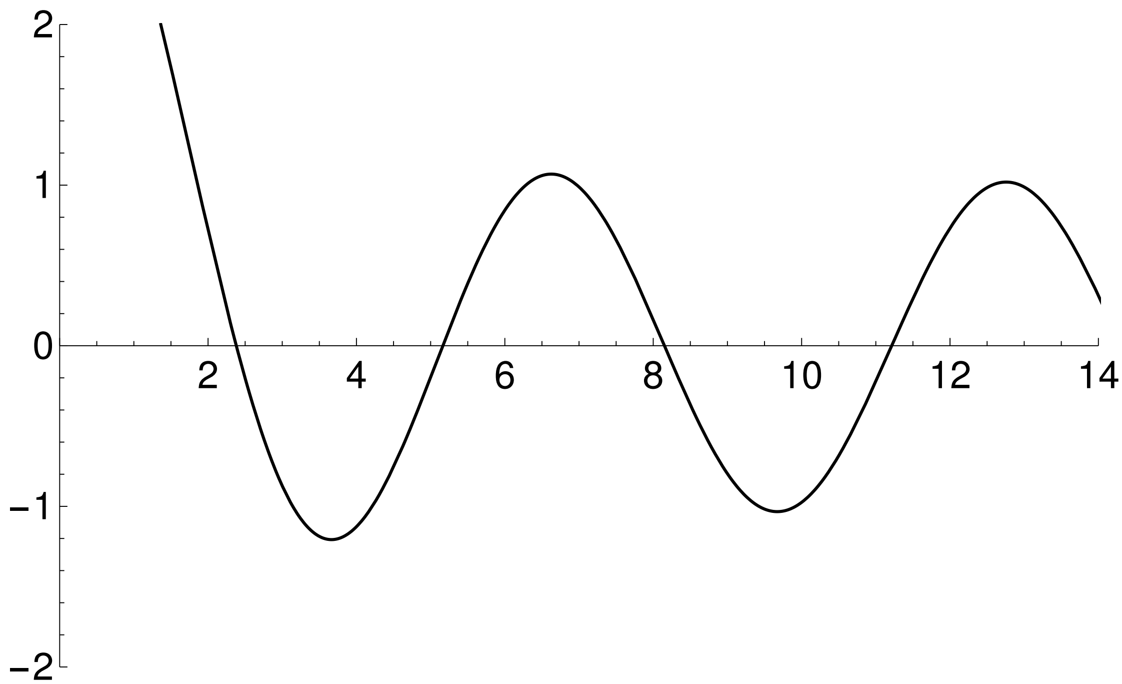
$\mathrm{Cos}[q] + \frac{\mathrm{epsilon}\ \mathrm{Sin}[q]}{2\,q}$

```
epsilon := 5

plus[k_] := 1

minus[k_] := -1
```

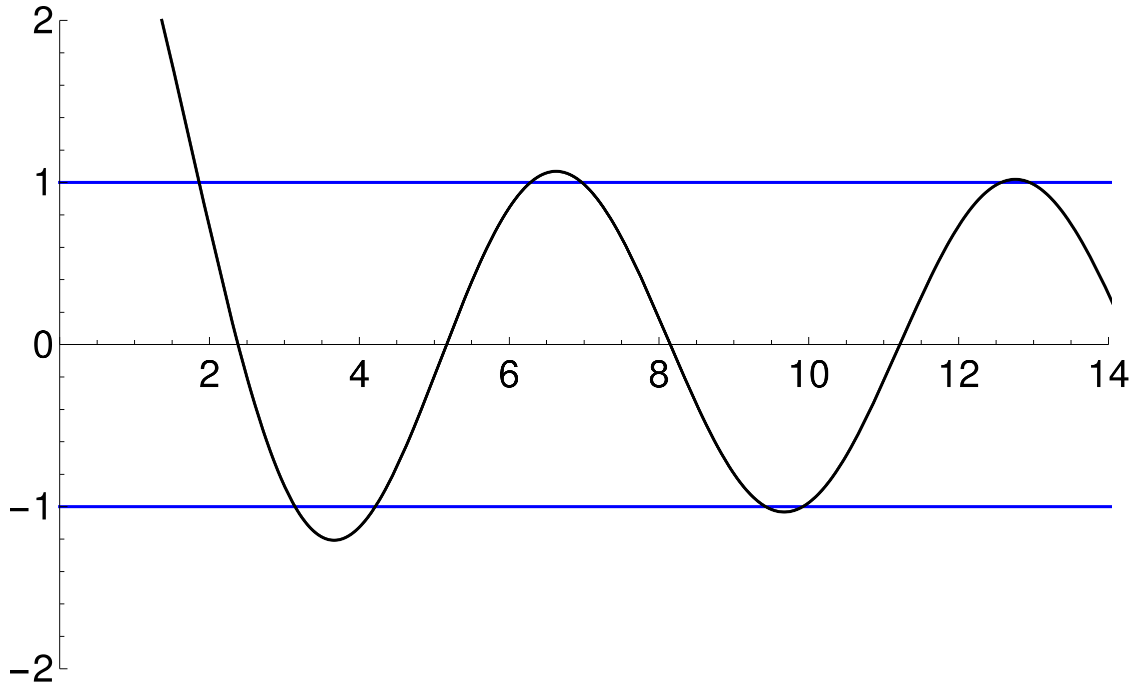
```
pones = Plot[{plus[k], minus[k]}, {k, 0, 19},
  PlotRange → {{0, 14}, {-2, 2}}, PlotStyle → Blue, AxesStyle → Directive[20]]
```



```
pf = Plot[f[k], {k, 0, 19}, PlotRange → {{0, 14}, {-2, 2}},
  PlotStyle → Black, AxesStyle → Directive[20]]
```



```
cosk = Show[pones, pf]
```




```
Export["cosk.png", cosk, ImageResolution → 300]
```

cosk.png