Numerical matrix method for quantum periodic potentials

Felipe Le Vot, Juan J. Meléndez and Santos B. Yuste

Departamento de Física and Instituto de Computación Científica Avanzada (ICCAEX), Universidad de Extremadura, 06006 Badajoz, Spain

Mathematica version...

```
In[1]:= StringJoin[{"The running Mathematica version is ", ToString[$Version]}]
Out[1]= The running Mathematica version is
        10.1.0  for Microsoft Windows (64-bit) (March 24, 2015)
```

Dimerized KP Model

Functions that for a KP potential with lattice parameter a=1 and n_b barriers:
(I) evaluate the Hamiltonian matrix H_{nm} [Eq. (17)]
(II) solve Eq. (10) to find the eigenenergies and eigenfunctions.
The size of the system is $L=n_b*a=n_b$.
The positions of the nb barriers are stored in the vector (or list) xrV, and the corresponding widths and heights are stored in vectors bV and V0V, respectively.

Note that these vectors xrV, bV and V0V describe the periodic potential.
We use units such as $\hbar^2/2\mu$=1, μ being the mass of the particle.

Functions F_{nm} [Eq. (20)]

```
In[2]:= f[k_, x_, L_] := f[k, x, L] = Sin[k * Pi * x / L] / (Pi * k)
```

```
In[3]:= Fnm[n_, m_, x_, L_] := f[m - n, x, L] - f[m + n, x, L]
```

```
In[4]:= Fnn[n_, x_, L_] := x / L - f[2 * n, x, L]
```

Functions h_{nm} [Eq. (18)]

```
In[5]:= hnm[n_, m_, s_, b_, L_] := Fnm[n, m, s + b / 2, L] - Fnm[n, m, s - b / 2, L]
```

```
In[6]:= hnn[n_, s_, b_, L_] := Fnn[n, s + b / 2, L] - Fnn[n, s - b / 2, L]
```

Matrix elements H_{nm} [Eq. (17)]

In[7]:=
```
HHnn[n_, L_, V0V_, xrV_, bV_] := N[(n * Pi / L) ^2 +
    Sum[V0V[[k]] * hnn[n,  xrV[[k]], bV[[k]], L], {k, 1, Length[xrV]}] ]
```

In[8]:=
```
HHnm[n_, m_, L_, V0V_, xrV_, bV_] :=
 Sum[V0V[[k]] * hnm[n, m,  xrV[[k]], bV[[k]], L], {k, 1, Length[xrV]}] // N
```

The function *HamiltonianMatrix* provides the $N \times N$ Hamiltonian matrix for the KP potential given by xrV, bV and V0V, where N=Nterms

In[9]:=
```
HamiltonianMatrix[Nterms_, L_, V0V_, xrV_, bV_] :=
 Table[If[n == m, HHnn[n, L, V0V,  xrV, bV], HHnm[n, m, L, V0V,  xrV, bV]],
  {n, 1, Nterms}, {m, 1, Nterms}]
```

The function *soluKP* provides the solution of Eq. (13).
The list of energies is provided directly by the function *EnV*. The n-th element of this matrix is the eigenvalue E_n
The matrix of coefficients $c_m^{(n)}$ of Eq. (14) are provided by *cnMatrix*. The element (n,m) of this matrix is just the coefficient $c_m^{(n)}$.

In[10]:=
```
soluKP[Nterms_, L_, V0V_, xrV_, bV_] := soluKP[Nterms, L, V0V, xrV, bV] =
  Eigensystem[N[HamiltonianMatrix[Nterms, L, V0V, xrV, bV]]] // Chop //
     Transpose // Sort // Transpose
```

In[11]:=
```
EnV[Nterms_, L_, V0V_, xrV_, bV_] :=
 EnV[Nterms, L, V0V, xrV, bV] = soluKP[Nterms, L, V0V, xrV, bV][[1]]
```

In[12]:=
```
cnMatrix[Nterms_, L_, V0V_, xrV_, bV_] :=
 cnMatrix[Nterms, L, V0V, xrV, bV] = soluKP[Nterms, L, V0V, xrV, bV][[2]]
```

The function *psi* provides an estimate of n-th eigenfunction according the formula

$$\psi_n(x) \simeq \sum_{m=1}^{N^*} c_m^{(n)} \varphi_m(x)$$

When Nast=N^* is equal to Nterm=N , this expression is just Eq. (14).
The function ProbDensity is just $|\psi_n(x)|^2$

In[13]:=
```
psi[n_, x_, Nast_, Nterms_, L_, V0V_, xrV_, bV_] := Sqrt[2 / L] *
  Sum[cnMatrix[Nterms, L, V0V, xrV, bV][[n, m]] * Sin[m * Pi * x / L], {m, 1, Nast}]
```

In[14]:=
```
ProbDensity[n_, x_, Nast_, Nterms_, L_, V0V_, xrV_, bV_] :=
 Abs[psi[n, x, Nast, Nterms, L, V0V, xrV, bV]]^2
```

Definition of a small dimerized KP system and plot of the potential

We generate the list of positions, widths and heights of the nb barriers of the KP potential. In this *example* the nb=10 barriers have width q=1/6, height v0=100 and they are placed at the middle of each unit cell. The shift parameter is u=0.1

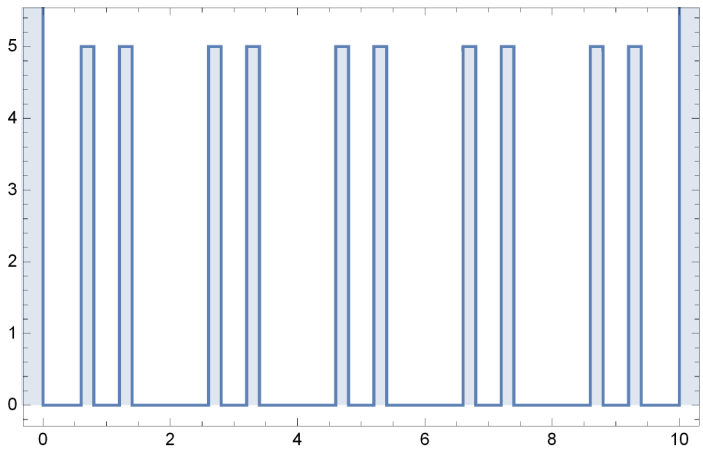
In[15]:=
```
nb = 10; L = N[nb]; q = 1. / 5; v0 = 1 / q; u = 0.2;
{xrVector = Table[-1. / 2 + k + u * (-1) ^ (k + 1), {k, 1, nb}],
 bVector = Table[q, {k, 1, nb}],
 V0Vector = Table[v0, {k, 1, nb}]}
```

Out[16]=
```
{{0.7, 1.3, 2.7, 3.3, 4.7, 5.3, 6.7, 7.3, 8.7, 9.3},
 {0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2},
 {5., 5., 5., 5., 5., 5., 5., 5., 5., 5.}}
```

The next instruction generates a plot of the potential defined above

In[17]:=
```
leftWall = Plot[If[x < 0, 1.2 * v0, 0],
    {x, -1, 0.01}, Filling → Bottom, PlotPoints → 100, Frame → True];
rightWall = Plot[If[x > L, 1.2 * v0, 0], {x, L - 0.01, L + 1},
  Filling → Bottom, PlotPoints → 100, Frame → True];
KPpotential = Plot[Piecewise[Table[{V0Vector[[k]],
      xrVector[[k]] - bVector[[k]] / 2 <= x <= xrVector[[k]] + bVector[[k]] / 2},
    {k, 1, nb}]], {x, 0, nb}, Filling → Bottom, PlotPoints → 2000, Frame → True];
Show[leftWall, rightWall, KPpotential, PlotRange → {{-0.1, L + 0.1}, {0, 1.05 v0}}]
```

Out[18]=



Evaluation of the first three gaps for the example and parameters considered in Fig. 6 (b=1/100, V_0=100, n_b=80,

N=Nterms=200) when u=0.2

Definition of the dimerized KP system and obtention of E_n

We generate the list of positions, widths and heights of the nb barriers of the KP potential.
In this *example* the nb=80, b=q=1/100, v0=100, *N*=Nterms=200 and u=0.2

In[19]:=
```
nb = 80; L = N[nb]; q = 1. / 100; v0 = 1 / q; u = 0.2;
{xrVector = Table[-1. / 2 + k + u * (-1) ^ (k + 1), {k, 1, nb}],
  bVector = Table[q, {k, 1, nb}],
  V0Vector = Table[v0, {k, 1, nb}]};
```

The vector of the first N=Nterms energies E_n is calculated by means of the function EnV for u=0.2

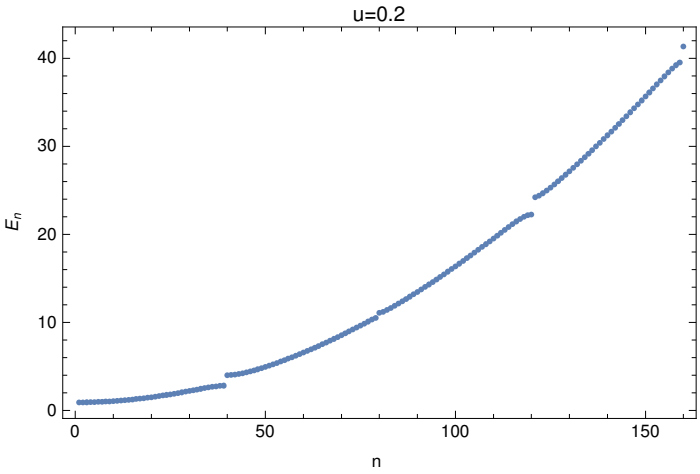
In[21]:=
```
Nterms = 200; Energies = EnV[Nterms, L, V0Vector, xrVector, bVector]
```

Out[21]=
```
{0.92555, 0.930041, 0.937524, 0.947998, 0.96146, 0.977908, 0.997337, 1.01974,
 1.04512, 1.07346, 1.10476, 1.139, 1.17618, 1.21628, 1.25928, 1.30518, 1.35394,
 1.40554, 1.45995, 1.51714, 1.57706, 1.63967, 1.70488, 1.77264, 1.84284,
 1.91536, 1.99004, 2.0667, 2.14506, 2.22479, 2.30544, 2.38638, 2.46677,
 2.54542, 2.62066, 2.69017, 2.75081, 2.7986, 2.82921, 4.02399, 4.05017, 4.0898,
 4.15275, 4.2347, 4.33166, 4.44049, 4.55887, 4.68514, 4.8181, 4.9569, 5.10092,
 5.24972, 5.40293, 5.56032, 5.72167, 5.88683, 6.05567, 6.2281, 6.40403,
 6.5834, 6.76615, 6.95222, 7.14159, 7.3342, 7.53002, 7.72902, 7.93116, 8.1364,
 8.34471, 8.55602, 8.77027, 8.98737, 9.20717, 9.42944, 9.6538, 9.87949,
 10.1048, 10.3253, 10.5243, 11.1002, 11.2058, 11.4141, 11.65, 11.897, 12.1505,
 12.4088, 12.6712, 12.9374, 13.207, 13.48, 13.7562, 14.0355, 14.318, 14.6036,
 14.8922, 15.1839, 15.4785, 15.776, 16.0765, 16.3799, 16.686, 16.9949,
 17.3066, 17.6208, 17.9376, 18.2568, 18.5783, 18.9018, 19.227, 19.5536, 19.881,
 20.2085, 20.5349, 20.8583, 21.1758, 21.482, 21.7678, 22.0154, 22.1934,
 22.2588, 24.2374, 24.4242, 24.6867, 24.9935, 25.327, 25.6778, 26.0407,
 26.4128, 26.7921, 27.1775, 27.5682, 27.9636, 28.3635, 28.7674, 29.1752,
 29.5866, 30.0017, 30.4202, 30.8421, 31.2672, 31.6956, 32.1271, 32.5617,
 32.9993, 33.4398, 33.8831, 34.3291, 34.7777, 35.2287, 35.6818, 36.1367,
 36.5929, 37.0496, 37.5054, 37.9584, 38.4044, 38.8353, 39.2322, 39.5478,
 41.3653, 41.5003, 41.8252, 42.2378, 42.6905, 43.1647, 43.652, 44.1486, 44.6521,
 45.1615, 45.6758, 46.1947, 46.7177, 47.2446, 47.7753, 48.3096, 48.8474,
 49.3887, 49.9334, 50.4814, 51.0327, 51.5872, 52.1451, 52.7061, 53.2704,
 53.8379, 54.4085, 54.9824, 55.5594, 56.1395, 56.7228, 57.3093, 57.8989,
 58.4916, 59.0875, 59.6865, 60.2886, 60.8938, 61.5022, 62.1137, 62.7133}
```

Plot of the energíes E_n obtained by means of the matrix method versus *n* for u=0.2

In[22]:=
```
ListPlot[Table[{n, Energies[[n]]}, {n, 1, 2 * nb}], Frame → True,
 FrameLabel -> {"n ", "En"}, PlotLabel → StringJoin[{"u=", ToString[u]}]]
```

Out[22]=



Evaluation of the gaps

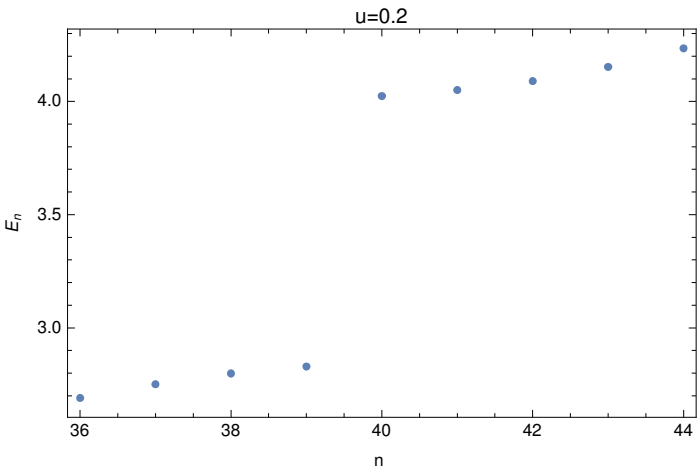
First gap

A zoom around the *first* gap and energies of the near levels for u=0.2

In[23]:=
```
zoom = Table[{n, Energies[[n]]}, {n, nb / 2 - 4, nb / 2 + 4}];
Print[{"n ", "En"}, "=", zoom]; ListPlot[zoom, Frame → True,
 FrameLabel -> {"n ", "En"}, PlotLabel → StringJoin[{"u=", ToString[u]}]]
```

```
{n , En}={{36, 2.69017}, {37, 2.75081}, {38, 2.7986}, {39, 2.82921},
  {40, 4.02399}, {41, 4.05017}, {42, 4.0898}, {43, 4.15275}, {44, 4.2347}}
```

Out[23]=



A crude estimation of the *first* gap that appears between n=39≡*ngap* and n=40 for the sifth u=0.2

In[24]:=
```
ngap = 39; {StringJoin[{"u=", ToString[u]}],
 Energies[[ngap + 1]] - Energies[[ngap]]}
```

Out[24]=
```
{u=0.2, 1.19479}
```

The improved estimation of the *first* gap that appears between n=39≡*ngap* and n=40 for u=0.2, obtained as the difference between the extrapolation value at the middle point from the right, E_{n+1} - (E_{n+2}-E_{n+1})/2, and from the left, E_n - (E_{n+1}-E_{n-1})/2.

In[25]:=
```
ngap = 39;
ExtrapolationValueFromTheRight =
 Energies[[ngap + 1]] - (Energies[[ngap + 2]] - Energies[[ngap + 1]]) / 2;
ExtrapolationValueFromTheLeft =
 Energies[[ngap]] + (Energies[[ngap]] - Energies[[ngap - 1]]) / 2;
Print[StringJoin[{"Extrapolation value from the right=",
    ToString[ExtrapolationValueFromTheRight]}]];
Print[StringJoin[{"Extrapolation value from the left=",
    ToString[ExtrapolationValueFromTheLeft]}]];
ImprovedValueOfTheGap = ExtrapolationValueFromTheRight -
  ExtrapolationValueFromTheLeft
```

```
Extrapolation value from the right=4.0109
Extrapolation value from the left=2.84451
```

Out[26]= 1.16639

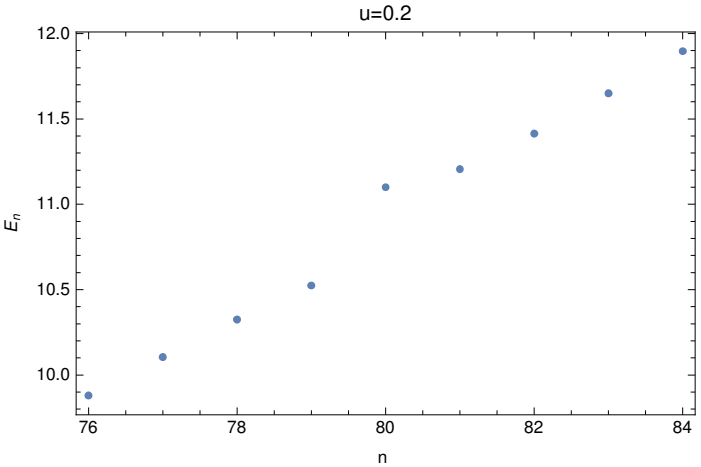
Second gap

A zoom around the *second* gap and energies of the near levels for u=0.2

In[27]:=
```
zoom = Table[{n, Energies[[n]]}, {n, nb - 4, nb + 4}];
Print[{"n ", "E_n"}, "=", zoom]; ListPlot[zoom, Frame → True,
 FrameLabel -> {"n ", "E_n"}, PlotLabel → StringJoin[{"u=", ToString[u]}]]
```

```
{n , E_n}={{76, 9.87949}, {77, 10.1048}, {78, 10.3253}, {79, 10.5243},
  {80, 11.1002}, {81, 11.2058}, {82, 11.4141}, {83, 11.65}, {84, 11.897}}
```

Out[27]=



A crude estimation of the *second* gap that appears between n=79≡*ngap* and n=80 for u=0.2

In[28]:=
```
ngap = 79; Energies[[ngap + 1]] - Energies[[ngap]]
```

Out[28]= 0.575834

The improved estimation of the *second* gap that appears between n=79≡*ngap* and n=80 for u=0.2, obtained as the difference between the extrapolation value at the middle point from the right, E_{n+1} - (E_{n+2}-E_{n+1})/2, and from the left, E_n - (E_{n+1}-E_{n-1})/2.

In[29]:=
```
ngap = 79;
ExtrapolationValueFromTheRight =
 Energies[[ngap + 1]] - (Energies[[ngap + 2]] - Energies[[ngap + 1]]) / 2;
ExtrapolationValueFromTheLeft =
 Energies[[ngap]] + (Energies[[ngap]] - Energies[[ngap - 1]]) / 2;
Print[StringJoin[{"Extrapolation value from the right=",
    ToString[ExtrapolationValueFromTheRight]}]];
Print[StringJoin[{"Extrapolation value from the left=",
    ToString[ExtrapolationValueFromTheLeft]}]];
{StringJoin[{"u=", ToString[u]}], ImprovedValueOfTheGap =
  ExtrapolationValueFromTheRight - ExtrapolationValueFromTheLeft}
```

```
Extrapolation value from the right=11.0473
Extrapolation value from the left=10.6239
```

Out[30]= {u=0.2, 0.423478}

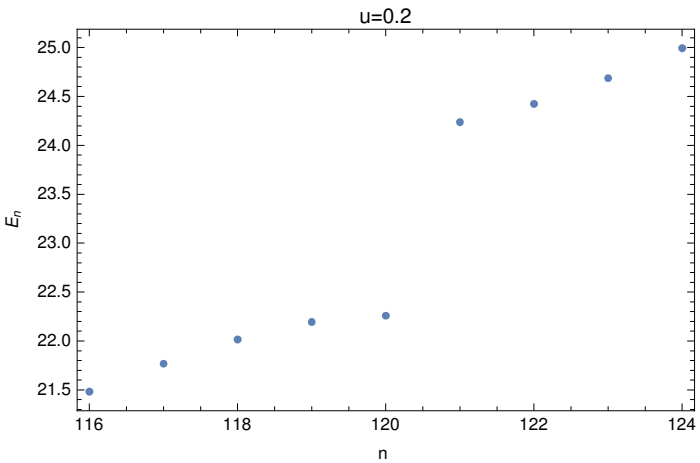
Third gap

A zoom around the *third* gap and energies of the near levels for u=0.2

In[31]:=
```
zoom = Table[{n, Energies[[n]]}, {n, 3 nb / 2 - 4, 3 nb / 2 + 4}];
Print[{"n ", "E_n"}, "=", zoom]; ListPlot[zoom, Frame → True,
 FrameLabel -> {"n ", "E_n"}, PlotLabel → StringJoin[{"u=", ToString[u]}]]
```

```
{n , E_n}={{116, 21.482}, {117, 21.7678}, {118, 22.0154}, {119, 22.1934},
  {120, 22.2588}, {121, 24.2374}, {122, 24.4242}, {123, 24.6867}, {124, 24.9935}}
```

Out[31]=



A crude estimation of the *third* gap that appears between n=120≡*ngap* and n=121 for u=0.2

In[32]:=
```
ngap = 120; Energies[[ngap + 1]] - Energies[[ngap]]
```

Out[32]= 1.97866

The improved estimation of the *third* gap that appears between n=120≡*ngap* and n=121 for u=0.2, obtained as the difference between the extrapolation value at the middle point from the right, E_{n+1} - (E_{n+2}-E_{n+1})/2, and from the left, E_n - (E_{n+1}-E_{n-1})/2.

In[33]:=
```
ngap = 120;
ExtrapolationValueFromTheRight =
 Energies[[ngap + 1]] - (Energies[[ngap + 2]] - Energies[[ngap + 1]]) / 2;
ExtrapolationValueFromTheLeft =
 Energies[[ngap]] + (Energies[[ngap]] - Energies[[ngap - 1]]) / 2;
Print[StringJoin[{"Extrapolation value from the right=",
   ToString[ExtrapolationValueFromTheRight]}]];
Print[StringJoin[{"Extrapolation value from the left=",
   ToString[ExtrapolationValueFromTheLeft]}]];
{StringJoin[{"u=", ToString[u]}], ImprovedValueOfTheGap =
  ExtrapolationValueFromTheRight - ExtrapolationValueFromTheLeft}
```

```
Extrapolation value from the right=24.144
Extrapolation value from the left=22.2914
```

Out[34]= {u=0.2, 1.8526}