

Молекула H2 гармоническая

Для всяких проверок правильности операций численного расчета энергий и волновых функций гармонического осциллятора написана программа H2parabolaA.m. Численные расчеты в этом случае не требуются для работы, поскольку для этого осциллятора известны аналитические решения одномерного стационарного уравнения Шредингера. Но для ангармонических потенциалов я буду решать задачи только численным методом, строя программу для потенциала Гука-Гаусса H2GG.m по образцу программы H2parabolaA.m.

Задан параболический потенциал такой, чтобы получалась частота перехода 4400 см^{-1} . При этом потенциал параметризован (k_q) так, чтобы колебательные координаты можно было бы выразить в Å.

Вот текст программы H2parabolaA.m.

```
% H2 with parabolic U
% x and q in Angstroms

global E kq k

mp = 1.67243e-24;
mv = mp/2; % reduced mass of H2 in q-vibr model
h = 6.62e-27; % *1e16 Then I may set distances in Angstroms
a = 0.4; % in Angstroms

nius = 4400;
niu = nius * 2.99793e10; % niu = 1.3191e+14 s^-1
v = 1;
Ev = h*niu*(v + 0.5);
kq = mv*4*pi^2*niu^2/1e16;; % U = 0.5*kq*q^2; Then
distances in Angstroms
dkq = 0.0001679e+5/1e16;
kq = kq + dkq;
k = 4*pi^2*2*mv/(h^2*1e16);
dE = 0.0;
E = Ev + dE;

x0 = 0;
xf = a;
if mod(v,2) ~= 0
    psi0 = [0 1];
else
    psi0 = [1 0];
end % if

[x1, Psi1] = ode23('waveh2p', [x0 xf], psi0);

xf = -a;
```

```

[x2, Psi2] = ode23('waveh2p', [x0 xf], psi0);

nx = length(x2);
x = zeros(nx-1,1);
Psi = zeros(nx-1,2);
for i = 1:nx-1
    x(i) = x2(nx -i +1);
    Psi(i,:) = Psi2(nx -i +1,:);
end % i

x = [x;x1];
Psi = [Psi;Psi1];
Psi_2 = Psi(:,1).^2;

An = trapz(x, Psi_2);
An = sqrt(An);
Psi = Psi/An;
Psi_2 = Psi(:,1).^2;
IntegrPsi_2 = trapz(x, Psi_2);
plot(x,Psi)
grid
% plot(x,Psi(:,1))
% grid

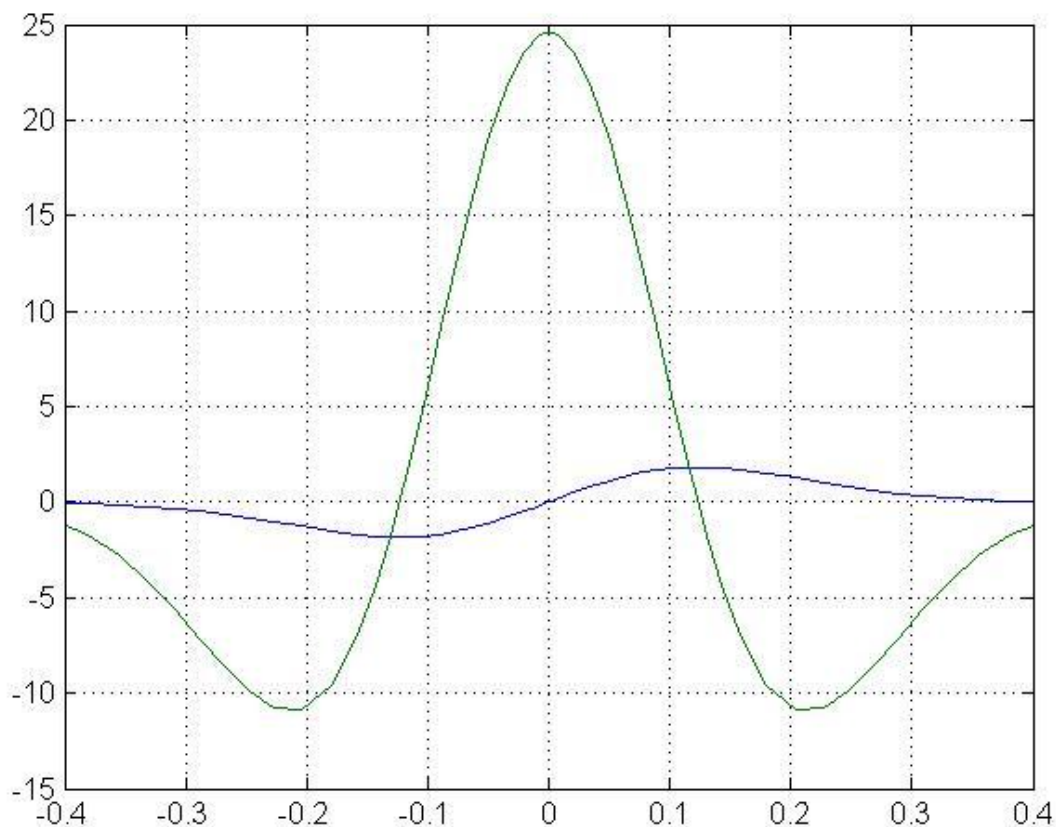
% U = (0.5*kq*x.^2)*1e13;
% E1 = E*1e13;
% hold on
% plot(x,U)
% plot([x(1) x(end)], [E1 E1], 'r')
% plot(x, Psi_2+E1, 'g')
% grid
% hold off
%
```

Некоторые участки текста программы закомментированы. Это связано с возможностями программы выводить на графики различные формы решения задачи. При необходимости можно декомментировать нужные участки и закомментировать ненужные. Для различных значений колебательного квантового числа U нужно изменять в тексте программы значение параметра a – полуширины области определения волновой функции. Параметр δkq менять не следует. Это подгоночный параметр, который подбирался один раз для того, чтобы решения уравнения Шредингера получались «хорошими». То есть для того, чтобы функция и ее производные сходились к нулю на границе области определения. Возможно, это потребовалось из-за свойств МатЛаб функции `ode23`, использованной для решения уравнения с заданными начальными условиями. Возможно, функция `ode45` работает в этой задаче лучше.

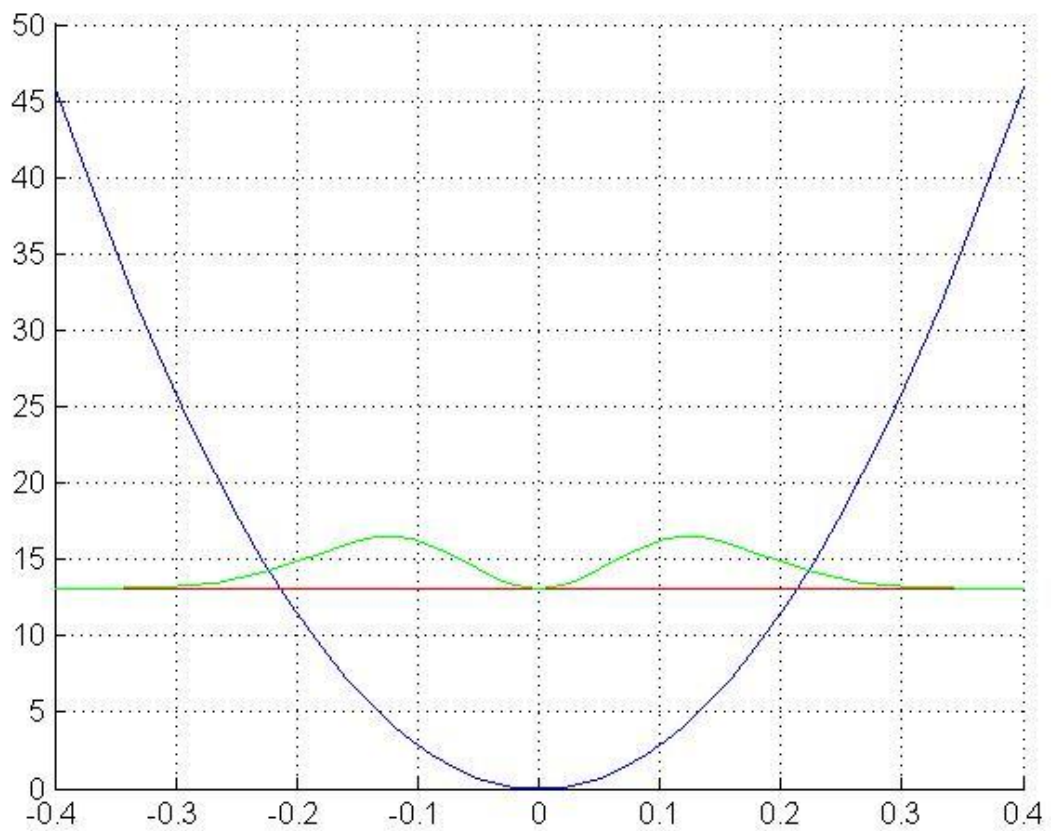
Программа обращается к функции `waveh2p.m`, вычисляющей абсциссы x и ординаты Ψ искомой волновой функции. Это требуется по правилам обращения к МатЛаб функции `ode23`. Вот текст этой программы-функции.

```
function xdot = waveh2p(x,psi)
global E kq k
xdot = [psi(2); -k*(E - 0.5*kq.*x.^2).*psi(1)];
```

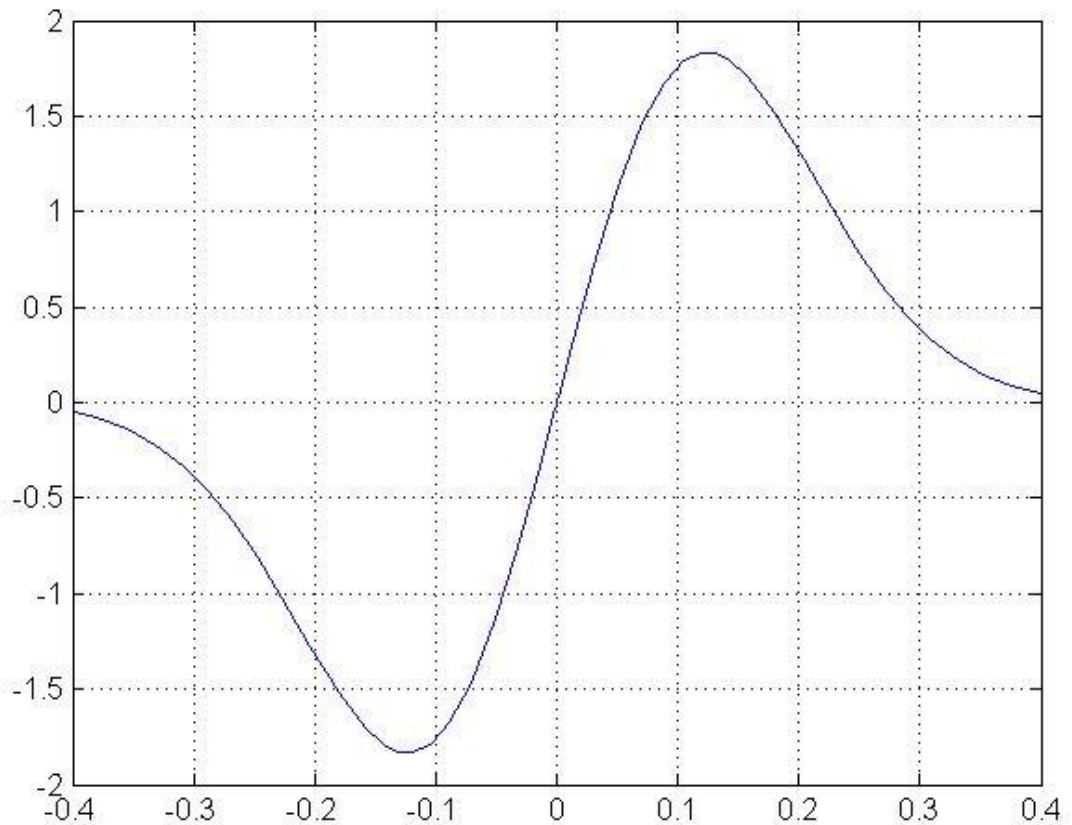
Численно решена квантовая задача для колебательного состояния с $\nu = 1$. Получена правильная энергия и правильная волновая функция, как показано на графиках.



Нормированная волновая функция (синий график) и ее производная (зеленый график) для $\nu = 1$. Численное решение.



Энергия $E = E_n = 1.3098555756 \times 10^{-12}$ эрг и квадрат волновой функции (зеленый график) для $n = 1$. Численное решение.

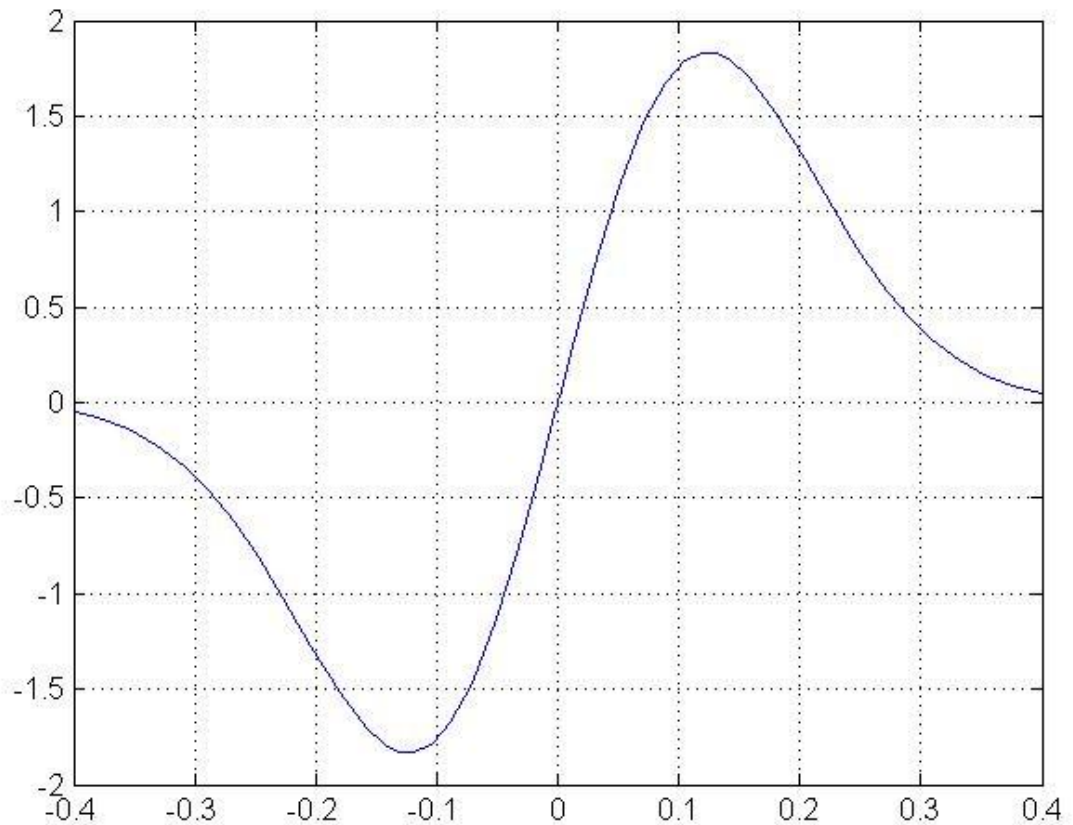


Отдельно подробно показана нормированная волновая функция для $\nu = 1$. Численное решение.

Сравним этот результат с теоретической волновой функцией, имеющей для гармонического осциллятора известное выражение

$$\Psi = \text{const} \exp(-0.5\beta^2 q^2) H_\nu(\beta q).$$

Согласно Ельяшевичу, и это особо отмечено в качестве его заслуги в предисловии к моей МатЛаб книге, $\beta = 1/q_{00}$, где q_{00} есть классическая амплитуда колебаний при $\nu = 0$. В программе эти величины рассчитываются. $q_{00} = 0.1233 \text{ \AA}$. Это согласуется со значением, приведенным в книге Ельяшевича. $\beta = 8.1106 \text{ \AA}^{-1}$. Для $\nu = 1$ $H_\nu(q) = \beta q$. Строим для тех же $x = q$, для которых построена численная волновая функция, теоретическую волновую функцию. Нормируем и выводим график. Он ничем не отличается от графика численной функции:



Теоретическая волновая функция для $\nu = 1$.

Следовательно, у меня есть аппарат для построения теоретических волновых функций гармонического осциллятора в форме таблиц и графиков. Это мне пригодится в работах с ангармоническими осцилляторами.

Явный вид полиномов Эрмита для различных ν .

$$H_0(x) = 1$$

$$H_1(x) = x$$

$$H_2(x) = x^2 - 1$$

$$H_3(x) = x^3 - 3x$$

$$H_4(x) = x^4 - 6x^2 + 3$$

$$H_5(x) = x^5 - 10x^3 + 15x$$

$$H_6(x) = x^6 - 15x^4 + 45x^2 - 15$$

$$H_7(x) = x^7 - 21x^5 + 105x^3 - 105x$$

$$H_8(x) = x^8 - 28x^6 + 210x^4 - 420x^2 + 105$$

Однако я не пользуюсь явными выражениями для полиномов Эрмита. У меня имеется функция chh.m, которая подсчитывает значения полинома для заданного u и массива абсцисс $\eta = \beta \cdot q$.

```
function h = chh(n,eta)

pmax=fix(n/2);

h = 0;

for p=0:pmax

    h = h + (-1)^p*eta.^(n-2*p)/(gamma(1+n-2*p)*gamma(1+p)*2^(2*p));

end

h=h*(2^n)*gamma(1+n);
```

Этой функцией я пользуюсь в программе psi_try.m, которая используется в дальнейшем, чтобы сравнивать решения одномерной ангармонической задачи о колебаниях молекулы водорода по программе H2GG.m с теоретическими аналитическими решениями для гармонического потенциала. Теоретическая волновая функция для $u = 1$ также получена с помощью этой программы независимо от программы H2GG.m при раскомментированном участке, определяющем параметры осциллятора.

```
% psi_try

% run after H2GG

psiav = Psi(:,1);

xa = x;

plot(xa,psiav)

grid

% mp = 1.67244e-24;

% mv = mp/2; % reduced mass of H2 in q-vibr model

% h = 6.62e-27; % *1e16 Then I may set distances in Angstroms

%

% nius = 4400; % niu spectroscopic

% niu = nius * 2.99793e10; %= 1.3191e+14 s-1

% kq = mv*4*pi^2*niu^2/1e16; % U = 0.5*kq*q^2; Then distances in Angstroms

%

% v = 5;
```

```
% Ev = h*niu*(v + 0.5);
```

```
q = -a:0.01:b;
```

```
q00 = sqrt(h*niu/kq);
```

```
beta = 1/q00;
```

```
eta = beta*q;
```

```
psihv=chh(v,eta).*exp(-0.5.*eta.^2);
```

```
psihv2 = psihv.^2;
```

```
Anteor = trapz(q, psihv2);
```

```
Anteor = sqrt(Anteor);
```

```
psihv = psihv/Anteor;
```

```
plot(q,psihv)
```

```
grid
```

```
% plot(x,Psi(:,1))
```

```
% grid
```

```
qlleft = -sqrt(2*Ev/kq);
```

```
qright = -qlleft;
```

```
qnew = q;
```

```
nq = length(q);
```

```
for i = 1:nq
```

```
    qnew(i) = q(i)*(qlleft/qleft + (xright/qright - qlleft/qleft)* ...
```

```
        (q(i) - qlleft)/(qright - qlleft) );
```

```
end % i
```

```
plot(qnew,psihv)
```

```
grid
```