QHD-2 Zero Point Motion in Energy Exchange between Two Oscillators

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx

Introduction

Second Order Quantized Hamilton Dynamics (QHD-2) is applied to the **energy exchange between two harmonic oscillators coupled by an anharmonic term**. QHD-2 gives an approximation to the Heisenberg Equations of Motion (EOM). The QHD commands used in this document are included in QUANTUM, which is a free *Mathematica* add-on that can be downloaded from
http://homepage.cem.itesm.mx/lgomez/quantum/
This tutorial shows how to use the QUANTUM *Mathematica* add-on to reproduce the results and graphs from Prezhdo and Pereverzev in J. Chem. Phys. Vol 113 No. 16, October 2000, Pages 6557-6565
http://homepage.cem.itesm.mx/lgomez/quantum/QHD2000.pdf
.

Load the Package

First load the Quantum`QHD` package. Write:
Needs["Quantum`QHD`"]
then press at the same time the keys SHIFT-ENTER to evaluate. *Mathematica* will load the package and print a welcome message:

```
Needs["Quantum`QHD`"]
```

```
Quantum`QHD`
A Mathematica package for Quantized Hamilton
  Dynamics approximation to Heisenberg Equations of Motion
by José Luis Gómez-Muñoz
based on the original idea of Kirill Igumenshchev

This add-on does NOT work properly with the debugger turned on. Therefore
  the debugger must NOT be checked in the Evaluation menu of Mathematica.

Execute SetQHDAliases[] in order to use the keyboard to enter QHD objects
SetQHDAliases[] must be executed again in each new notebook that is created
```

In order to use the keyboard to enter quantum objects write:
SetQHDAliases[ ]
then press at the same time the keys SHIFT-ENTER to evaluate. Remember that SetQuantumAliases[ ] must be evaluated again in each

new notebook:

```
SetQHDAliases[]
```

```
ALIASES:
[ESC]on[ESC]      · Quantum concatenation symbol
[ESC]time[ESC]    t Time symbol
[ESC]hb[ESC]      ℏ Reduced Planck's constant (h bar)
[ESC]ii[ESC]      ⅈ Imaginary I symbol
[ESC]inf[ESC]     ∞ Infinity symbol
[ESC]->[ESC]      → Option (Rule) symbol
[ESC]ave[ESC]     ⟨□⟩ Quantum average template
[ESC]expec[ESC]   ⟨□⟩ Quantum average template
[ESC]symm[ESC]    (□·□)ₛ Symmetrized quantum product template
[ESC]comm[ESC]    ⟦□,□⟧₋ Commutator template
[ESC]po[ESC]      (□)^□ Power template
[ESC]su[ESC]      □_□ Subscripted variable template
[ESC]posu[ESC]    □_□^□ Power of a subscripted variable template
                  □
[ESC]fra[ESC]     ── Fraction template
                  □
[ESC]eva[ESC]     □/.{□→□,□→□} Evaluation (ReplaceAll) template

SetQHDAliases[] must be executed again in each
  new notebook that is created, only one time per notebook.
```

Commutation Relationships and Hamiltonian J. Chem. Phys. Vol 113 No. 16, October 2000, Pages 6557-6565

The Hamiltoninan and commutation relationships that are defined below are those used by Prezhdo and Pereverzev in J. Chem. Phys. Vol 113 No. 16, October 2000, Pages 6557-6565
http://homepage.cem.itesm.mx/lgomez/quantum/QHD2000.pdf. In order to enter the templates and symbols $\square_{\square}$, $\square_{\square}^{\square}$, $⟦\square, \square⟧_{-}$, ω, λ and i you can either use the QHD palette (toolbar) or press the keys [ESC]su[ESC], [ESC]supo[ESC], [ESC]comm[ESC], [ESC]omega[ESC], [ESC]lambda[ESC] and [ESC]ii[ESC]

```
SetQuantumObject[q, p];
[[q1, p1]]- = i;
[[q2, p2]]- = i;
[[q1, p2]]- = 0;
[[q2, p1]]- = 0;
[[q1, q2]]- = 0;
[[p1, p2]]- = 0;
zph = p1^2/2 + (ω1^2 * q1^2)/2 + p2^2/2 + (ω2^2 * q2^2)/2 + λ * q1 · q2^2
```

$$\frac{p_1^2}{2}+\frac{p_2^2}{2}+\frac{1}{2}\,q_1^2\,\omega_1^2+\frac{1}{2}\,q_2^2\,\omega_2^2+\lambda\,q_1\cdot q_2^2$$

Heisenberg Equations of Motion and Cross Terms Approximation

The evolution of the expected value of an observable *A* in the Heisenberg representation is given by the equation of motion (EOM):

$$i\hbar\frac{d}{dt}\langle A\rangle=\langle[A,\,H]\rangle$$

The Heisenberg EOM can be calculated using the QHD-*Mathematica* command QHDEOM, as shown below. The first argument specifies the QHD-order, which is set below to ∞ so that no QHD closure is made in this first example. The second argument is the list of observables. In this example we are intereted in those obsevables needed to calculate the harmonic energies, $\{p_1^2, p_2^2, q_1^2, q_1^2\}$. The third argument is the Hamiltonian, remember that it was stored in variable *zph* above in this document. Finally options can be included, for example QHDHBar→1 specifies that reduced Planck's constant (Dirac constant) will take the value of 1 (The → symbol can be entered by pressing the keys [ESC]->[ESC], that is [ESC][MINUS][GREATERTHAN][ESC]). The output of QHDEOM can be stored in a variable; it is stored in *zeom*, then it is shown in a nice format with the command QHDForm below:

```
zeom = QHDEOM[∞, {p1^2, p2^2, q1^2, q2^2}, zph, QHDHBar → 1];
QHDForm[zeom]
```

No closure was applied
QHDApproximantFunction->QHDCrossTermsApproximant.

$$\frac{d\langle p_1^2\rangle}{dt}=-2\,\lambda\,\langle p_1\rangle\,\langle q_2^2\rangle-2\,\omega_1^2\,\langle p_1 q_1\rangle_s$$

$$\frac{d\langle p_2^2\rangle}{dt}=-2\,\omega_2^2\,\langle p_2 q_2\rangle_s-4\,\lambda\,\langle q_1\rangle\,\langle p_2 q_2\rangle_s$$

$$\frac{d\langle q_1^2\rangle}{dt}=2\,\langle p_1 q_1\rangle_s$$

$$\frac{d\langle q_2^2\rangle}{dt}=2\,\langle p_2 q_2\rangle_s$$

Notice that the EOM contain new expectation values, for example $\langle p_1\cdot q_1\rangle_s=\left\langle\frac{p_1\cdot q_1+q_1\cdot p_1}{2}\right\rangle$

The label above the EOM specifies that QHDCrossTermsApproximant was used as an approximation. The definition of this approximation is shown below:

```
? QHDCrossTermsApproximant
```

QHDCrossTermsApproximant[expr] approximates expected values of crossterms $\langle p_1^n \cdot p_2^m \rangle \approx \langle p_1^n \rangle \langle p_2^m \rangle$ in expr

We can get the Heisenberg EOM without approximation using QHDApproximantFunction→Identity as the last argument of QHDEOM, as shown below:

```
zeo2 = QHDEOM[∞, {p1^2, p2^2, q1^2, q2^2}, zph,
    QHDHBar → 1, QHDApproximantFunction → Identity];
QHDForm[
 zeo2]
```

No closure was applied

$$\frac{d\langle p_1^2\rangle}{dt} = -2\,\omega_1^2\,\langle p_1 q_1\rangle_s - 2\,\lambda\,\langle p_1 q_2^2\rangle_s$$

$$\frac{d\langle p_2^2\rangle}{dt} = -2\,\omega_2^2\,\langle p_2 q_2\rangle_s - 4\,\lambda\,\langle p_2 q_1 q_2\rangle_s$$

$$\frac{d\langle q_1^2\rangle}{dt} = 2\,\langle p_1 q_1\rangle_s$$

$$\frac{d\langle q_2^2\rangle}{dt} = 2\,\langle p_2 q_2\rangle_s$$

Products of operators were symmetrized, for example: $\langle p_2 \cdot q_1 \cdot q_2\rangle_s = \frac{1}{2}\langle p_2 \cdot q_1 \cdot q_2\rangle + \frac{1}{2}\langle q_2 \cdot q_1 \cdot p_2\rangle$

QHD-2 Closure Approximation

Equations of motion (EOM) were obtained above for those variables, $\{p_1^2, p_2^2, q_1^2, q_1^2\}$, that are needed for calculation of the harmonic energies. Those calculations are by default performed with cross-term approximations $\langle p_1 \cdot p_2\rangle \approx \langle p_1\rangle\langle p_2\rangle$, and it is possible to recalculate the EOM without approximation using QHDApproximantFunction→Identity. In any case, new expected values like $\langle p_1\, q_1\rangle_s$ are included in those EOM. If the EOM of those new expected values is calculated, more expected values will be generated, building an infinite hierarchy of EOM. Therefore, a closure procedure is applied in order to have a finite hierarchy of EOM. For instance, the approximation:

$\langle ABC\rangle \approx \langle AB\rangle\langle C\rangle + \langle AC\rangle\langle B\rangle + \langle BC\rangle\langle A\rangle - 2\langle A\rangle\langle B\rangle\langle C\rangle$

is used to approximate third-order expected values like $\langle q^3\rangle$ and $\langle pq^2\rangle_s$ in terms of the firts and second order expected values $\langle p\rangle$, $\langle q\rangle$, $\langle p^2\rangle$, $\langle q^2\rangle$ and $\langle p \cdot q\rangle_s = \left\langle \frac{p\cdot q + q\cdot p}{2}\right\rangle$. Such a QHD-2 hierarchy is obtained below as the output of the command QHDHierarchy with the first argument set to 2. Same as above, the second argument is the initial list of expected values in the hierarchy, the third argument is the hamiltonian, and optional arguments can be given after that. The output of QHDHierarchy is stored in the variable *zphier*, and it is shown using the QHDForm command below:

```
zphier = QHDHierarchy[2, {p1^2, p2^2, q1^2, q2^2}, zph, QHDHBar → 1];
QHDForm[ zphier ]
```

Closure procedure was applied to order 2
QHDApproximantFunction->QHDCrossTermsApproximant.

$$\frac{d\langle p_1^2\rangle}{dt} = -2\lambda\langle p_1\rangle\left\langle q_2^2\right\rangle - 2\omega_1^2\langle p_1 q_1\rangle_s$$

$$\frac{d\langle p_2^2\rangle}{dt} = -2\omega_2^2\langle p_2 q_2\rangle_s - 4\lambda\langle q_1\rangle\langle p_2 q_2\rangle_s$$

$$\frac{d\langle q_1^2\rangle}{dt} = 2\langle p_1 q_1\rangle_s$$

$$\frac{d\langle q_2^2\rangle}{dt} = 2\langle p_2 q_2\rangle_s$$

$$\frac{d\langle p_1\rangle}{dt} = -\omega_1^2\langle q_1\rangle - \lambda\left\langle q_2^2\right\rangle$$

$$\frac{d\langle p_1 q_1\rangle_s}{dt} = \left\langle p_1^2\right\rangle - \omega_1^2\left\langle q_1^2\right\rangle - \lambda\langle q_1\rangle\left\langle q_2^2\right\rangle$$

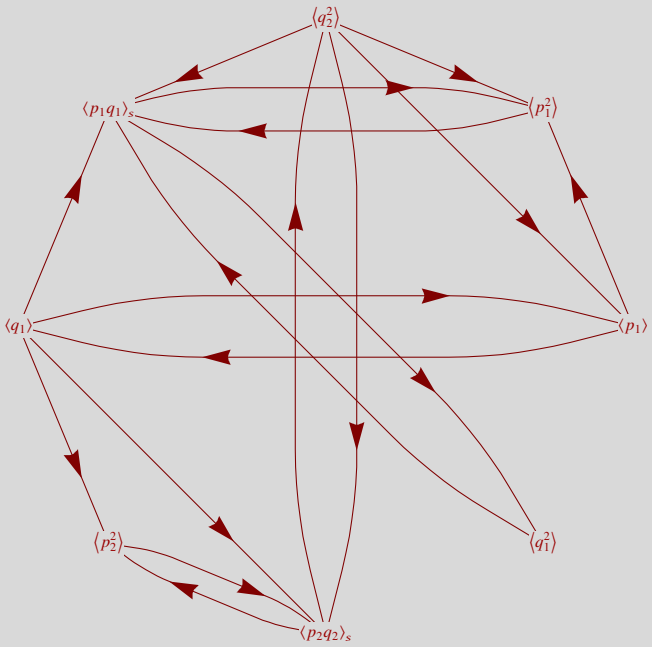
$$\frac{d\langle q_1\rangle}{dt} = \langle p_1\rangle$$

$$\frac{d\langle p_2 q_2\rangle_s}{dt} = \left\langle p_2^2\right\rangle - \omega_2^2\left\langle q_2^2\right\rangle - 2\lambda\langle q_1\rangle\left\langle q_2^2\right\rangle$$

A EOM hierarchy can be shown in a graph using the command QHDGraphPlot on the output of QHDHierachy. Each arrow points from a first dynamical variable to a second dynamical variable that includes the first one in its EOM; please compare the table above with the graph below:

```
QHDGraphPlot[zphier]
```

Closure procedure was applied to order 2
QHDApproximantFunction->QHDCrossTermsApproximant.



The command QHDDifferentialEquations formats the output of QHDHierarchy as standard *Mathematica* equations, as those that can be part of the input of standard *Mathematica* commands like DSolve and NDSolve. The output *zphier* that was stored above is formated that way below:

```
QHDDifferentialEquations[zphier]
```

```
{⟨p1²⟩'[t] == -2 λ ⟨p1⟩[t] ⟨q2²⟩[t] - 2 ω1² ⟨(p1 · q1)s⟩[t],
 ⟨p2²⟩'[t] == -2 ω2² ⟨(p2 · q2)s⟩[t] - 4 λ ⟨q1⟩[t] ⟨(p2 · q2)s⟩[t], ⟨q1²⟩'[t] == 2 ⟨(p1 · q1)s⟩[t],
 ⟨q2²⟩'[t] == 2 ⟨(p2 · q2)s⟩[t], ⟨p1⟩'[t] == -ω1² ⟨q1⟩[t] - λ ⟨q2²⟩[t],
 ⟨(p1 · q1)s⟩'[t] == ⟨p1²⟩[t] - ω1² ⟨q1²⟩[t] - λ ⟨q1⟩[t] ⟨q2²⟩[t], ⟨q1⟩'[t] == ⟨p1⟩[t],
 ⟨(p2 · q2)s⟩'[t] == ⟨p2²⟩[t] - ω2² ⟨q2²⟩[t] - 2 λ ⟨q1⟩[t] ⟨q2²⟩[t]}
```

A different symbol for time can be specified. The operator → can be entered pressing the keys [ESC][MINUS][GREATERTHAN][ESC], see the equations below:

```
QHDDifferentialEquations[zphier, QHDSymbolForTime → z]
```

$$\left\{\left\langle p_1^2\right\rangle'[z] == -2\,\lambda\,\langle p_1\rangle[z]\,\left\langle q_2^2\right\rangle[z] - 2\,\omega_1^2\,\langle (p_1\cdot q_1)_s\rangle[z],\right.$$
$$\left\langle p_2^2\right\rangle'[z] == -2\,\omega_2^2\,\langle (p_2\cdot q_2)_s\rangle[z] - 4\,\lambda\,\langle q_1\rangle[z]\,\langle (p_2\cdot q_2)_s\rangle[z],$$
$$\left\langle q_1^2\right\rangle'[z] == 2\,\langle (p_1\cdot q_1)_s\rangle[z],\ \left\langle q_2^2\right\rangle'[z] == 2\,\langle (p_2\cdot q_2)_s\rangle[z],$$
$$\langle p_1\rangle'[z] == -\omega_1^2\,\langle q_1\rangle[z] - \lambda\,\left\langle q_2^2\right\rangle[z],$$
$$\langle (p_1\cdot q_1)_s\rangle'[z] == \left\langle p_1^2\right\rangle[z] - \omega_1^2\,\left\langle q_1^2\right\rangle[z] - \lambda\,\langle q_1\rangle[z]\,\left\langle q_2^2\right\rangle[z],\ \langle q_1\rangle'[z] == \langle p_1\rangle[z],$$
$$\left.\langle (p_2\cdot q_2)_s\rangle'[z] == \left\langle p_2^2\right\rangle[z] - \omega_2^2\,\left\langle q_2^2\right\rangle[z] - 2\,\lambda\,\langle q_1\rangle[z]\,\left\langle q_2^2\right\rangle[z]\right\}$$

Solution of the QHD-2 Equations: Zero Point Energy

Next we evaluate the hierarchy with the numerical values of the parameters that were used by Prezhdo and Pereverzev in their paper. The result of that evaluation is stored in the variable *zpnumhier*:

```
zpnumhier = zphier /. {ω1 → 1.0, ω2 → 1.1, λ → -0.11};
QHDForm[zpnumhier]
```

Closure procedure was applied to order 2
QHDApproximantFunction->QHDCrossTermsApproximant.

$\frac{d\langle p_1^2\rangle}{dt} = 0.22\,\langle p_1\rangle\,\left\langle q_2^2\right\rangle - 2.\,\langle p_1 q_1\rangle_s$
$\frac{d\langle p_2^2\rangle}{dt} = -2.42\,\langle p_2 q_2\rangle_s + 0.44\,\langle q_1\rangle\,\langle p_2 q_2\rangle_s$
$\frac{d\langle q_1^2\rangle}{dt} = 2\,\langle p_1 q_1\rangle_s$
$\frac{d\langle q_2^2\rangle}{dt} = 2\,\langle p_2 q_2\rangle_s$
$\frac{d\langle p_1\rangle}{dt} = -1.\,\langle q_1\rangle + 0.11\,\left\langle q_2^2\right\rangle$
$\frac{d\langle p_1 q_1\rangle_s}{dt} = \left\langle p_1^2\right\rangle - 1.\,\left\langle q_1^2\right\rangle + 0.11\,\langle q_1\rangle\,\left\langle q_2^2\right\rangle$
$\frac{d\langle q_1\rangle}{dt} = \langle p_1\rangle$
$\frac{d\langle p_2 q_2\rangle_s}{dt} = \left\langle p_2^2\right\rangle - 1.21\,\left\langle q_2^2\right\rangle + 0.22\,\langle q_1\rangle\,\left\langle q_2^2\right\rangle$

The command QHDInitialConditionsTemplate generates an initial conditions template for the hierarchy:

```
QHDInitialConditionsTemplate[ zpnumhier, 0 ]
```

$$\left\{\left\langle p_1^2\right\rangle[0] == \blacksquare,\ \left\langle p_2^2\right\rangle[0] == \blacksquare,\ \left\langle q_1^2\right\rangle[0] == \blacksquare,\ \left\langle q_2^2\right\rangle[0] == \blacksquare,\right.$$
$$\left.\langle p_1\rangle[0] == \blacksquare,\ \langle (p_1\cdot q_1)_s\rangle[0] == \blacksquare,\ \langle q_1\rangle[0] == \blacksquare,\ \langle (p_2\cdot q_2)_s\rangle[0] == \blacksquare\right\}$$

Copy-paste the output of the previous command as input in the next one. Fill in the placeholders (■) with the appropiate initial values. These initial conditions are used in J. Chem. Phys. Vol 113 No. 16, October 2000, Pages 6557-6565
http://homepage.cem.itesm.mx/lgomez/quantum/QHD2000.pdf

The initial conditions are stored in the variable *zpinicond* below:

```
zpinicond = {⟨p1^2⟩[0] == 0.5, ⟨p2^2⟩[0] == 0.5, ⟨q1^2⟩[0] == 0.5, ⟨q2^2⟩[0] == 0.5,
   ⟨p1⟩[0] == 0.0, ⟨(p1 · q1)s⟩[0] == 0.0, ⟨q1⟩[0] == 0.0, ⟨(p2 · q2)s⟩[0] == 0.0}
```

```
{⟨p1^2⟩[0] == 0.5, ⟨p2^2⟩[0] == 0.5, ⟨q1^2⟩[0] == 0.5, ⟨q2^2⟩[0] == 0.5,
 ⟨p1⟩[0] == 0., ⟨(p1 · q1)s⟩[0] == 0., ⟨q1⟩[0] == 0., ⟨(p2 · q2)s⟩[0] == 0.}
```

The command QHDNDSolve takes as arguments the numerical version of the hierarchy (which was stored in the variable *zpnumhier*), the initial conditions (which were stored in *zpinicond*), the initial time and the final time. The output of this command is the numerical solution of the differential equations, in the form of InterpolatingFunction objects. This output can be used to plot (graph) any function of the dynamical variables (expected values), as it will be shown below in this document. The output is stored in the variable *zpsol* in the calculation below:

```
zpsol = QHDNDSolve[zpnumhier, zpinicond, 0, 50]
```

```
{{⟨p1^2⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨p2^2⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨q1^2⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨q2^2⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨p1⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨(p1 · q1)s⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨q1⟩ → InterpolatingFunction[{{0., 50.}}, <>],
  ⟨(p2 · q2)s⟩ → InterpolatingFunction[{{0., 50.}}, <>]}}
```

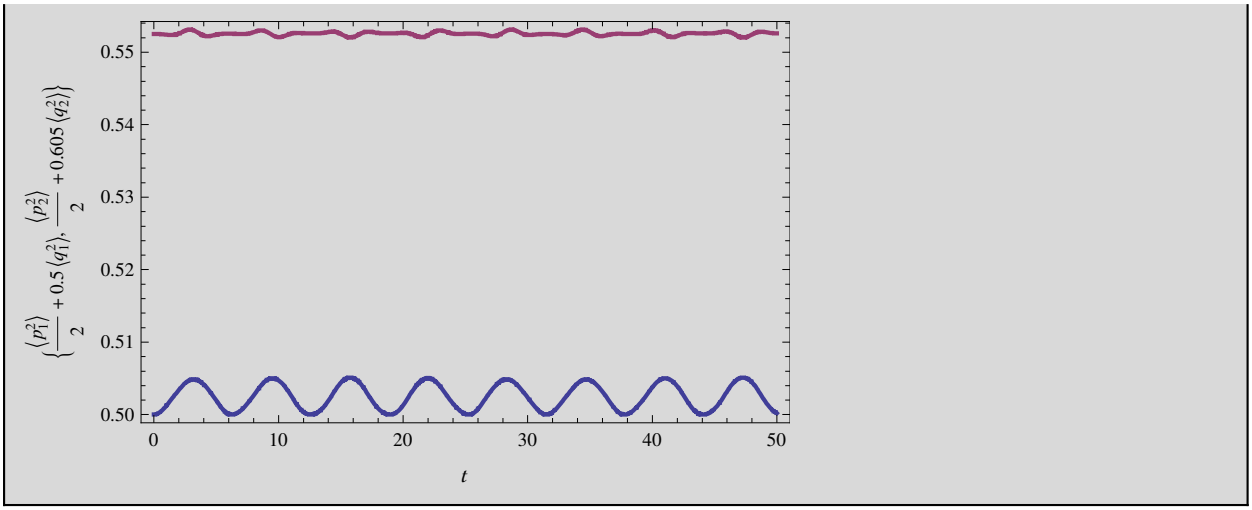
The expression for the harmonic energies of the system are stored in the list *energies* below:

```
energies = {p1^2/2 + ω1^2 q1^2/2, p2^2/2 + ω2^2 q2^2/2} /. {ω1 → 1.0, ω2 → 1.1}
```

```
{p1^2/2 + 0.5 q1^2, p2^2/2 + 0.605 q2^2}
```

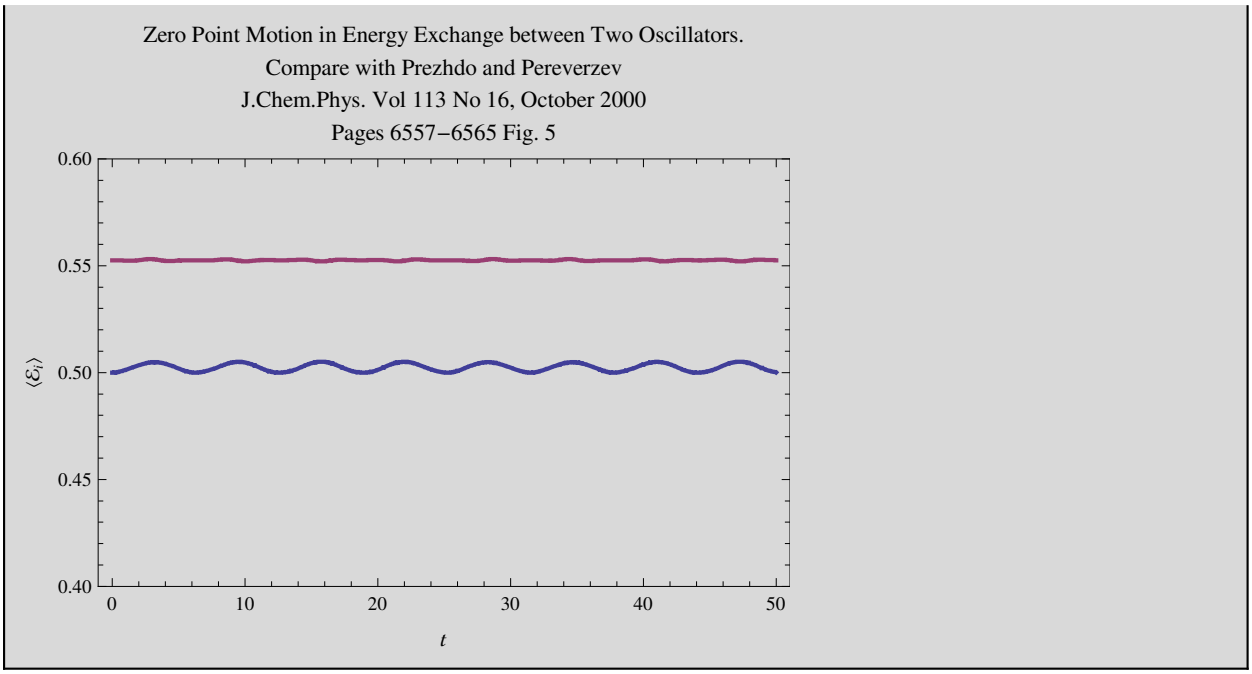
The command command QHDPlot takes as its second argument the output of QHDNDSolve, which was stored in the variable *zpsol*. The first argument specifies the expression that we want to plot (graph) as a function of time. The two harmonic energies are shown below:

```
QHDPlot[energies, zpsol]
```



QHDPlot accepts the same options as the standard *Mathematica* command Plot. Some of those options are used to reproduce part of a graph from Prezhdo and Pereverzev, see below:

```
QHDPlot[energies, zpsol,
 PlotRange → {0.4, 0.6},
 FrameLabel → {t, HoldForm[⟨ℰi⟩]},
 PlotLabel →
  "Zero Point Motion in Energy Exchange between Two Oscillators.\nCompare
    with Prezhdo and Pereverzev\nJ.Chem.Phys. Vol
    113 No 16, October 2000\nPages 6557-6565 Fig. 5"]
```



by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx