Theory Computing Project Notebook

Tingyu Chen

School of Physics and Astronomy

University of Manchester

February 2025

Week 3 - Matrix methods:

Task 1: Simple Matrices:

$$Av=\lambda v$$

Now it is your turn. Use the built in functions to find the Eigenvectors and Eigenvalues of the matrix below

In[66]:=
```
myNewMatrix = {{2.0, -1.0}, {-1.0, 3.0}};
myNewMatrix // MatrixForm
               矩阵格式
```

Out[67]//MatrixForm=

$$\begin{pmatrix} 2. & -1. \\ -1. & 3. \end{pmatrix}$$

```
eigenVecs = Eigenvectors[myNewMatrix]
            特征向量
eigenVals = Eigenvalues[myNewMatrix]
            特征值
```

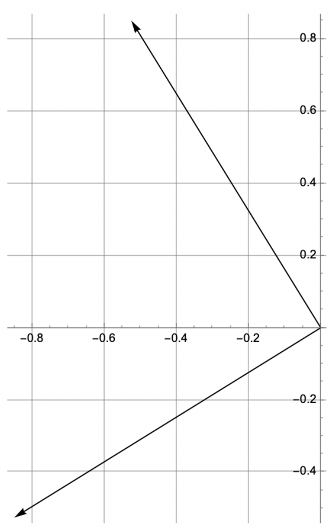
Out[75]= {{-0.525731, 0.850651}, {-0.850651, -0.525731}}

Out[76]= {3.61803, 1.38197}

Now plot those eigenvectors. Apply the matrix to the eigenvectors and see the eigenvalues make sense

In[83]:=
```
Graphics[{Arrow[{{0, 0}, eigenVecs[[1]]}], Arrow[{{0, 0}, eigenVecs[[2]]}]}, Axes → True, GridLines → Automatic]
图形      箭头                              箭头                                 坐标轴  真       网格线          自动
```

Out[83]=



In[106]:=
```
leftMatrix = myNewMatrix.eigenVecs[[1]]
rightMatrix = eigenVals[[1]] * eigenVecs[[1]]
leftMatrix == rightMatrix
```

Out[106]= {-1.90211, 3.07768}

Out[107]= {-1.90211, 3.07768}

Out[108]= True

We know that applying the matrix A to an eigenvector scales it by its eigenvalue:

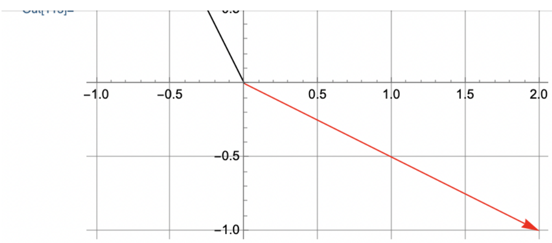
$$Av_i = \lambda_i v_i$$

Thus,

$$A \cdot \text{myNewVector} = A \cdot (av_1 + bv_2)$$

Since $Av_i = \lambda_i v_i$, this simplifies to:

$$A \cdot \text{myNewVector} = a\lambda_1 v_1 + b\lambda_2 v_2$$



We see that this vector changes direction under this matrix multiplication, and so is definitely not an eigenvector itself

```
In[116]:= coeffs = Solve[myNewVector == a * eigenVecs[[1]] + b * eigenVecs[[2]]]
                   解方程
          myNewVector // MatrixForm
                         矩阵格式
          a * eigenVecs[[1]] + b * eigenVecs[[2]] /. coeffs // MatrixForm
                                                               矩阵格式
Out[116]= {{a → 2.22703, b → -0.200811}}

Out[117]//MatrixForm=
          ( -1. )
          (  2. )

Out[118]//MatrixForm=
          ( -1.  2. )
```

The important realisation here is that by understanding the decomposition of any vector into eigenvectors, can understand the effect of the matrix on it. Stated another way, eigenvalues and eigenvectors contain all necessary information about the matrix. Try finding the result of matrix on vector above by using its eigen decomposition

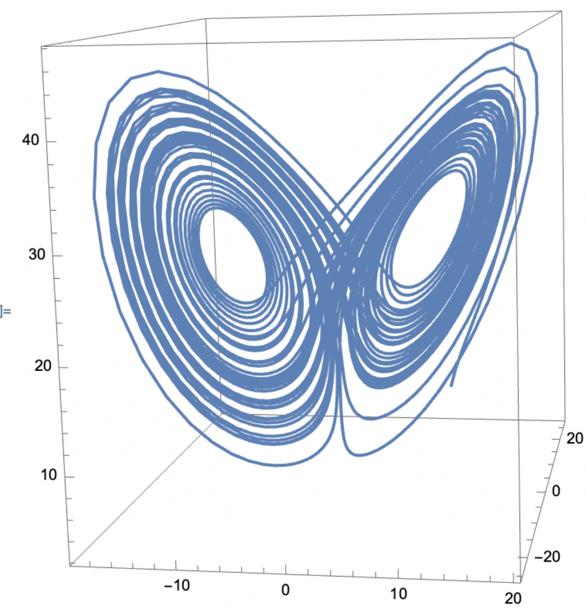
```
In[120]:= newVector = coeffs[[1, 1, 2]] * eigenVals[[1]] * eigenVecs[[1]] + coeffs[[1, 2, 2]] * eigenVals[[2]] * eigenVecs[[2]]
          directResult = myNewMatrix.myNewVector;
          newVector == directResult

Out[120]= {-4., 7.}

Out[122]= True
```

Task 2: Simple ODE:

For cases with no analytic solution we can use a numerical solution NDSolve

In[7]:=
```
(*Code 'borrowed' and slightly adapted from Wikipedia*)
tend = 50;
eq = {x'[t] == σ (y[t] - x[t]), y'[t] == x[t] (ρ - z[t]) - y[t], z'[t] == x[t] × y[t] - β z[t]};
init = {x[0] == 10, y[0] == 10, z[0] == 10};
pars = {σ → 10, ρ → 28, β → 8 / 3};
Sols = NDSolve[{eq /. pars, init}, {x, y, z}, {t, 0, tend}][[1]];
        数值求解微分方程组
ParametricPlot3D[{x[t], y[t], z[t]} /. Sols, {t, 0, tend}]
绘制三维参数图
```

Out[12]=



Task 3: Springs Masses:

We are interested in transforming our equations into the eigenbasis. Thankfully eigenanalysis is easy in Mathematica.

```
In[18]:= Eigsys = Eigensystem[ODEmatrix]
                  特征系统
```

Out[18]= $\left\{\left\{-\frac{3k}{m}, -\frac{k}{m}, 0\right\}, \{\{1, -2, 1\}, \{-1, 0, 1\}, \{1, 1, 1\}\}\right\}$

Make sure you can qualitatively explain the three eigenstates with their respective eigenvalue. Remember that the eigenvalues will be associated with the frequency of oscillation.

Construct the transformation matrices discussed in the notes. Remember that the *columns* of TmatrixInv should be the eigenvectors.

```
In[19]:= TmatrixInv = Transpose[Eigsys⟦2⟧];
                      转置
         Tmatrix = Inverse[TmatrixInv];
                   逆

         Tmatrix // MatrixForm
                    矩阵格式
         TmatrixInv // MatrixForm
                       矩阵格式
```

Out[21]//MatrixForm=

$$\begin{pmatrix} \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

Out[22]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 1 \\ -2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Essentially T will map our physical coordinates (x1, x2, x3) into the eigenstates, or normal modes (y1, y2, y3). T^{-1} does the opposite.
Show that T acting on X=(1,-2,1) indeed gives purely the first eigenstate. Similarly, show that the state Y=(0,1,0) gives X=(-1,0,1)

```
In[23]:= X1 = {1, -2, 1};
         Y1 = Tmatrix.X1;
         Y1 // MatrixForm
               矩阵格式
```

Out[25]//MatrixForm=

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

```
In[26]:= Y2 = {0, 1, 0};
         X2 = TmatrixInv.Y2;
         X2 // MatrixForm
               矩阵格式
```

Out[28]//MatrixForm=

$$\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Use the results above to construct the matrix
from TQT^{-1} where Q is the original ODE matrix.
Show it is the same as the diagonal matrix of eigenvalues

In[34]:=
```
diagonalmatrix = Tmatrix.ODEmatrix.TmatrixInv;
diagonalmatrix // MatrixForm
```
矩阵格式

Out[35]//MatrixForm=

$$\begin{pmatrix} -\frac{3k}{m} & 0 & 0 \\ 0 & -\frac{k}{m} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

In[31]:=
```
eigenVals = Eigsys[[1]]
DiagonalMatrixOfEig = DiagonalMatrix[eigenVals];

DiagonalMatrixOfEig // MatrixForm
```
对角矩阵 矩阵格式

Out[31]= $\left\{-\frac{3k}{m}, -\frac{k}{m}, 0\right\}$

Out[33]//MatrixForm=

$$\begin{pmatrix} -\frac{3k}{m} & 0 & 0 \\ 0 & -\frac{k}{m} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Let's print out each set of ODEs to show the reduction of complexity. Note however that even in the second case it would be simpler labelling as yi[t] rather than xi[t]

In[37]:=
```
For [i = 1, i ≤ Length[Statevec], i += 1,
    Print[D[Statevec[[i]], {t, 2}] == (ODEmatrix.Statevec)[[i]] + Avec[[i]]]
]
```
For循环 长度 打印 偏导

$$x1''[t] == -\frac{a\,k}{m} - \frac{k\,x1[t]}{m} + \frac{k\,x2[t]}{m}$$

$$x2''[t] == \frac{k\,x1[t]}{m} - \frac{2\,k\,x2[t]}{m} + \frac{k\,x3[t]}{m}$$

$$x3''[t] == \frac{a\,k}{m} + \frac{k\,x2[t]}{m} - \frac{k\,x3[t]}{m}$$

In[36]:=
```
For [i = 1, i ≤ Length[Statevec], i += 1,
  Print[D[(Tmatrix.Statevec)[[i]], {t, 2}] ==
    ((Tmatrix.ODEmatrix.TmatrixInv).(Tmatrix.Statevec))[[i]] + (Tmatrix.Avec)[[i]]]
]
```
For循环 长度 打印 偏导

$$\frac{x1''[t]}{6} - \frac{x2''[t]}{3} + \frac{x3''[t]}{6} == -\frac{3\,k\left(\frac{x1[t]}{6} - \frac{x2[t]}{3} + \frac{x3[t]}{6}\right)}{m}$$

$$-\frac{1}{2}\,x1''[t] + \frac{x3''[t]}{2} == \frac{a\,k}{m} - \frac{k\left(-\frac{x1[t]}{2} + \frac{x3[t]}{2}\right)}{m}$$

$$\frac{x1''[t]}{3} + \frac{x2''[t]}{3} + \frac{x3''[t]}{3} == 0$$

While the second form is clearly simpler due to its decoupling of equations, we will follow from the notes with yi'' = λi yi + bi (this is equivalent to what is written above)

```
In[38]:= Bvec = Tmatrix.Avec;
         Bvec // MatrixForm
```

Out[39]//MatrixForm=

$$\begin{pmatrix} 0 \\ \frac{a\,k}{m} \\ 0 \end{pmatrix}$$

```
In[40]:= Print[DSolve[y1''[t] == Eigsys[[1]][[1]] × y1[t] + Bvec[[1]], y1[t], t]]
```

$$\left\{\left\{y1[t] \to c_1 \operatorname{Cos}\left[\frac{\sqrt{3}\,\sqrt{k}\,t}{\sqrt{m}}\right] + c_2 \operatorname{Sin}\left[\frac{\sqrt{3}\,\sqrt{k}\,t}{\sqrt{m}}\right]\right\}\right\}$$

Now solve the other equations

```
In[41]:= For [i = 1, i ≤ Length[Statevec], i += 1,
          Print[DSolve[y''[t] == Eigsys[[1]][[i]] × y[t] + Bvec[[i]], y[t], t] /.
            y → ToExpression["y" <> ToString[i]]]
         ]
```

$$\left\{\left\{y1[t] \to c_1 \operatorname{Cos}\left[\frac{\sqrt{3}\,\sqrt{k}\,t}{\sqrt{m}}\right] + c_2 \operatorname{Sin}\left[\frac{\sqrt{3}\,\sqrt{k}\,t}{\sqrt{m}}\right]\right\}\right\}$$

$$\left\{\left\{y2[t] \to a + c_1 \operatorname{Cos}\left[\frac{\sqrt{k}\,t}{\sqrt{m}}\right] + c_2 \operatorname{Sin}\left[\frac{\sqrt{k}\,t}{\sqrt{m}}\right]\right\}\right\}$$

$$\{\{y3[t] \to c_1 + t\,c_2\}\}$$

Relate the qualitative understanding of eigenvectors to the results here.

```
Eigenvectors corresponds to the fundamental patterns of motion for this 3-mass system,
 the normal modes, and the Eigenvalues are the squared frequencies at which each pattern oscillates.
```

Task 4: QSHM:

For the Hermite Polynomials,

```
In[42]:= HermiteH[0, x]
         埃尔米特多项式
         HermiteH[1, x]
         埃尔米特多项式
         HermiteH[2, x]
         埃尔米特多项式
         HermiteH[3, x]
         埃尔米特多项式
```

Out[42]= 1

Out[43]= $2\,x$

Out[44]= $-2 + 4\,x^2$

Out[45]= $-12\,x + 8\,x^3$

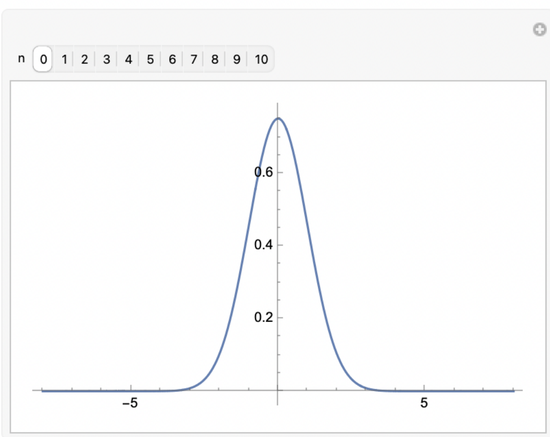
In[50]:= $\text{Psi}[n\_] := \frac{1}{\sqrt{2^n\ \text{Factorial}[n]}} \left(\frac{1}{\pi}\right)^{1/4} e^{-\frac{x^2}{2}}\ \text{HermiteH}[n, x]$

埃尔米特多项式

Let's check this looks sensible

```
In[51]:= Manipulate[
         交互式操作
           Plot[Psi[n], {x, -8, 8}, PlotRange → Full],
           绘图                     绘制范围     全范围
           {n, Range[0, 10]},
               范围
            ControlType → Setter}
            控件类型        设置按钮
         ]
```

Out[51]=



$\mathrm{H}\ \Psi\mathrm{n} = \mathrm{En}\ \Psi\mathrm{n}$

Use the function above to check this is indeed an eigenfunction H Ψn = En Ψn. Check the energy matches expectation. If you have a recent version of mathematica (≥ 12.3), the SolveValues function is nice, but otherwise the Solve function works fine.

```
In[53]:= H[psi_] := 1/2*(-D[psi, {x, 2}] + x^2 psi);
                        偏导
         eqn = H[Psi[n]] == En Psi[n];
         EneigenValue = SolveValues[eqn, En];
                        由解确定的值
         EneigenValue // FullSimplify
                         完全简化
```

Out[56]= $\left\{\frac{1}{2} + n\right\}$

Use the above idea to find the matrix for H. Think of HΨ as a new state Φ, then 'project' Φ back onto its corresponding Ψ amplitudes. If Ψn are eigenstates then this matrix will be diagonal.

```
In[71]:= nmax = 5;
         unperturbedHamiltonian = Table[Integrate[Psi[m] * H[Psi[n]], {x, -Infinity, Infinity}],
                                  表格  积分                               无穷大     无穷大
           {m, 0, nmax}, {n, 0, nmax}];
         FullSimplify[unperturbedHamiltonian]
         完全简化
         unperturbedHamiltonian // MatrixForm
                                   矩阵格式
```

Out[72]= $\left\{\left\{\frac{1}{2}, 0, 0, 0, 0, 0\right\}, \left\{0, \frac{3}{2}, 0, 0, 0, 0\right\}, \left\{0, 0, \frac{5}{2}, 0, 0, 0\right\}, \left\{0, 0, 0, \frac{7}{2}, 0, 0\right\}, \left\{0, 0, 0, 0, \frac{9}{2}, 0\right\}, \left\{0, 0, 0, 0, 0, \frac{11}{2}\right\}\right\}$

ut[73]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{5}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{7}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{9}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{11}{2} \end{pmatrix}$$

Use the above results to construct dH in a smarter way.
Hint: Use nested if statement or a switch statement

If [condition, result_if_true, result_if_false]

Note that the "result_if_false" can be another if statement.

In[127]:=
```
dH = Table[Table[
    λ If[m - n == -4, Coefficient[xhat4psi, psi[n - 4]],
      If[m - n == -2, Coefficient[xhat4psi, psi[n - 2]],
       If[m - n == 0, Coefficient[xhat4psi, psi[n]],
        If[m - n == 2, Coefficient[xhat4psi, psi[n + 2]],
         If[m - n == 4, Coefficient[xhat4psi, psi[n + 4]],
          0
          ]
         ]
        ]
       ]
      ],
    {n, 0, 10}],
   {m, 0, 10}];
```

In[128]:=
```
dH // MatrixForm
```

Out[128]//MatrixForm=

$$\begin{pmatrix}
\frac{3\lambda}{4} & 0 & \frac{3\lambda}{\sqrt{2}} & 0 & \sqrt{\frac{3}{2}}\,\lambda & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{15\lambda}{4} & 0 & 5\sqrt{\frac{3}{2}}\,\lambda & 0 & \sqrt{\frac{15}{2}}\,\lambda & 0 & 0 & 0 & 0 & 0 \\
\frac{3\lambda}{\sqrt{2}} & 0 & \frac{39\lambda}{4} & 0 & 7\sqrt{3}\,\lambda & 0 & 3\sqrt{\frac{5}{2}}\,\lambda & 0 & 0 & 0 & 0 \\
0 & 5\sqrt{\frac{3}{2}}\,\lambda & 0 & \frac{75\lambda}{4} & 0 & 9\sqrt{5}\,\lambda & 0 & \sqrt{\frac{105}{2}}\,\lambda & 0 & 0 & 0 \\
\sqrt{\frac{3}{2}}\,\lambda & 0 & 7\sqrt{3}\,\lambda & 0 & \frac{123\lambda}{4} & 0 & 11\sqrt{\frac{15}{2}}\,\lambda & 0 & \sqrt{105}\,\lambda & 0 & 0 \\
0 & \sqrt{\frac{15}{2}}\,\lambda & 0 & 9\sqrt{5}\,\lambda & 0 & \frac{183\lambda}{4} & 0 & 13\sqrt{\frac{21}{2}}\,\lambda & 0 & 3\sqrt{21}\,\lambda & 0 \\
0 & 0 & 3\sqrt{\frac{5}{2}}\,\lambda & 0 & 11\sqrt{\frac{15}{2}}\,\lambda & 0 & \frac{255\lambda}{4} & 0 & 15\sqrt{14}\,\lambda & 0 & 3\sqrt{35}\,\lambda \\
0 & 0 & 0 & \sqrt{\frac{105}{2}}\,\lambda & 0 & 13\sqrt{\frac{21}{2}}\,\lambda & 0 & \frac{339\lambda}{4} & 0 & 51\sqrt{2}\,\lambda & 0 \\
0 & 0 & 0 & 0 & \sqrt{105}\,\lambda & 0 & 15\sqrt{14}\,\lambda & 0 & \frac{435\lambda}{4} & 0 & 57\sqrt{\frac{5}{2}}\,\lambda \\
0 & 0 & 0 & 0 & 0 & 3\sqrt{21}\,\lambda & 0 & 51\sqrt{2}\,\lambda & 0 & \frac{543\lambda}{4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3\sqrt{35}\,\lambda & 0 & 57\sqrt{\frac{5}{2}}\,\lambda & 0 & \frac{663\lambda}{4}
\end{pmatrix}$$

Verify your method is correct!

In[129]:=
```
dH == dHBrute // MatrixForm
```

Out[129]//MatrixForm=

True

The unperturbed harmonic oscillator eigenstates have definite parity: even n give even functions and odd n give odd functions. The x^4 operator (and hence the perturbation dH) is an even function. An even operator only connects states with the same parity, if odd will be zero after integration. In the ground state (which is even), only even-indexed basis states mix in. Thus the coefficients for odd-indexed basis states remain very close to zero.

Now we construct the full matrix. Let's use a smarter way of finding H

In[130]:=
```
H = DiagonalMatrix[Table[(n + 1/2), {n, 0, Length[dH] - 1}]] + dH;
    对角矩阵           表格                   长度

H // MatrixForm
     矩阵格式
```

Out[131]//MatrixForm=

$$\begin{pmatrix}
\frac{1}{2}+\frac{3\lambda}{4} & 0 & \frac{3\lambda}{\sqrt{2}} & 0 & \sqrt{\frac{3}{2}}\,\lambda & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3}{2}+\frac{15\lambda}{4} & 0 & 5\sqrt{\frac{3}{2}}\,\lambda & 0 & \sqrt{\frac{15}{2}}\,\lambda & 0 & 0 & 0 & 0 & 0 \\
\frac{3\lambda}{\sqrt{2}} & 0 & \frac{5}{2}+\frac{39\lambda}{4} & 0 & 7\sqrt{3}\,\lambda & 0 & 3\sqrt{\frac{5}{2}}\,\lambda & 0 & 0 & 0 & 0 \\
0 & 5\sqrt{\frac{3}{2}}\,\lambda & 0 & \frac{7}{2}+\frac{75\lambda}{4} & 0 & 9\sqrt{5}\,\lambda & 0 & \sqrt{\frac{105}{2}}\,\lambda & 0 & 0 & 0 \\
\sqrt{\frac{3}{2}}\,\lambda & 0 & 7\sqrt{3}\,\lambda & 0 & \frac{9}{2}+\frac{123\lambda}{4} & 0 & 11\sqrt{\frac{15}{2}}\,\lambda & 0 & \sqrt{105}\,\lambda & 0 & 0 \\
0 & \sqrt{\frac{15}{2}}\,\lambda & 0 & 9\sqrt{5}\,\lambda & 0 & \frac{11}{2}+\frac{183\lambda}{4} & 0 & 13\sqrt{\frac{21}{2}}\,\lambda & 0 & 3\sqrt{21}\,\lambda & 0 \\
0 & 0 & 3\sqrt{\frac{5}{2}}\,\lambda & 0 & 11\sqrt{\frac{15}{2}}\,\lambda & 0 & \frac{13}{2}+\frac{255\lambda}{4} & 0 & 15\sqrt{14}\,\lambda & 0 & 3\sqrt{35}\,\lambda \\
0 & 0 & 0 & \sqrt{\frac{105}{2}}\,\lambda & 0 & 13\sqrt{\frac{21}{2}}\,\lambda & 0 & \frac{15}{2}+\frac{339\lambda}{4} & 0 & 51\sqrt{2}\,\lambda & 0 \\
0 & 0 & 0 & 0 & \sqrt{105}\,\lambda & 0 & 15\sqrt{14}\,\lambda & 0 & \frac{17}{2}+\frac{435\lambda}{4} & 0 & 57\sqrt{\frac{5}{2}}\,\lambda \\
0 & 0 & 0 & 0 & 0 & 3\sqrt{21}\,\lambda & 0 & 51\sqrt{2}\,\lambda & 0 & \frac{19}{2}+\frac{543\lambda}{4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3\sqrt{35}\,\lambda & 0 & 57\sqrt{\frac{5}{2}}\,\lambda & 0 & \frac{21}{2}+\frac{663\lambda}{4}
\end{pmatrix}$$

This Hamiltonian is not diagonal, but will be close to diagonal if λ is small. Let's find the diagonal version of this matrix by numerically finding the eigenvectors.
Use Eigensystem[...] and inspect the results

In[132]:=
```
physicalValues = {λ → 0.01};
eigsys = Eigensystem[H /. physicalValues] // N;
         特征系统                              数值

Print[eigsys[[1]]]
打印

Print[eigsys[[2]][[-1]]]
打印
```

{12.4642, 11.0616, 9.40909, 8.21787, 7.04873, 5.90126, 4.77491, 3.6711, 2.59085, 1.53565, 0.507256}

{0.999947, 5.32585×10^{-20}, -0.0100028, 1.59208×10^{-18}, -0.00257946, 3.56318×10^{-18}, 0.000187355, -6.59195×10^{-17}, 0.0000179496, 3.74253×10^{-17}, -4.24277×10^{-6}}

Tried it but failed, don't know why...

Use what we have learned in this notebook to make a plot of the groundstate energy as a function of λ.

In[169]:=
```
groundStateEnergy[lam_] :=
 Min[
  Eigenvalues[N[DiagonalMatrix[Table[n + 1/2, {n, 0, Length[dH] - 1}]] +
     (dH /. {λ → lam})]]] // N
groundStateEnergy[0.01]
Plot[Evaluate[groundStateEnergy[lam]], {lam, 0, 0.1},
 AxesLabel → {"λ", "Ground State Energy"}, PlotRange → All]
```

SetDelayed: $\left(\frac{1}{4}\text{Root}\left[13366080 + 676602432\lambda + 10552490928\lambda^2 + \ll 5\gg + (1953904 + 78727968\lambda + 878320872\,\text{Power}[\ll 2\gg] + 2876920200\,\text{Power}[\ll 2\gg] + 1895268375\,\text{Power}[\ll 2\gg])\#1^2 + (-163680 - 5211504\lambda - 40388040\,\text{Power}[\ll 2\gg] - 67151700\,\text{Power}[\ll 2\gg])\#1^3 + \ll 3\gg \&, 6\right]\right)$[lam_] 中的标签 Times 被保护.

Out[169]= $Failed

Out[170]=
$$\left(\frac{1}{4}\text{Root}\left[13\,366\,080 + 676\,602\,432\,\lambda + 10\,552\,490\,928\,\lambda^2 + 59\,445\,761\,760\,\lambda^3 + 109\,522\,759\,500\,\lambda^4 + 45\,218\,873\,700\,\lambda^5 + 1\,404\,728\,325\,\lambda^6 + \left(-9\,987\,648 - 470\,176\,608\,\lambda - 6\,582\,011\,040\,\lambda^2 - 31\,054\,106\,160\,\lambda^3 - 41\,229\,688\,500\,\lambda^4 - 8\,428\,369\,950\,\lambda^5\right)\#1 + \left(1\,953\,904 + 78\,727\,968\,\lambda + 878\,320\,872\,\lambda^2 + 2\,876\,920\,200\,\lambda^3 + 1\,895\,268\,375\,\lambda^4\right)\#1^2 + \left(-163\,680 - 5\,211\,504\,\lambda - 40\,388\,040\,\lambda^2 - 67\,151\,700\,\lambda^3\right)\#1^3 + \left(6700 + 148\,500\,\lambda + 592\,515\,\lambda^2\right)\#1^4 + (-132 - 1518\,\lambda)\#1^5 + \#1^6 \,\&,\, 6\right]\right)[0.01]$$

Out[171]=