In[1]:=
```
SetOptions[SelectedNotebook[],
 PrintingStyleEnvironment → "Printout", ShowSyntaxStyles → True]
```

In[2]:=
```
<< Notation`
```

In[3]:=
```
(* Simplify the notation for the
 formulas: index vectors based on a subscript notation *)
Notation[ x_n_ ⟺ x_⟦n_⟧ ]
```

In[4]:=
```
d = 1.0; (* Equilibrium bond length *)
α = Reverse@{13.00773, 1.962079, 0.444529, 0.1219492} & /@ Range[1, 2] // Flatten
```

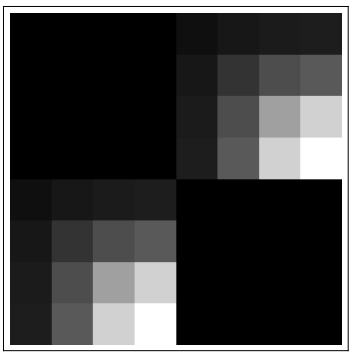
Out[5]=
```
{0.121949, 0.444529, 1.96208, 13.0077, 0.121949, 0.444529, 1.96208, 13.0077}
```

In[6]:=
```
P = {{0, 0, 0}, {d, 0, 0}}; (* Atomic positions *)
Z = Partition[Table[#, {n, 1, 4}] & /@ P // Flatten, 3]
 (* Array of positions for the coefficients α *)
```

Out[7]=
```
{{0, 0, 0}, {0, 0, 0}, {0, 0, 0}, {0, 0, 0},
 {1., 0, 0}, {1., 0, 0}, {1., 0, 0}, {1., 0, 0}}
```

In[8]:=
```
K = Array[Exp[-(α#1 α#2)/(α#1 + α#2) Norm[Z#1 - Z#2]^2] &, {8, 8}]; (* Coefficients matrix K_{p,q} *)
#[K] & /@ {SymmetricMatrixQ, ArrayPlot, MatrixForm}
```

Out[9]=

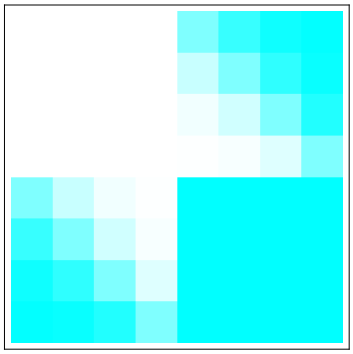
{True,  ,

$$\begin{pmatrix} 1. & 1. & 1. & 1. & 0.940847 \\ 1. & 1. & 1. & 1. & 0.90874 \\ 1. & 1. & 1. & 1. & 0.891533 \\ 1. & 1. & 1. & 1. & 0.886197 \\ 0.940847 & 0.90874 & 0.891533 & 0.886197 & 1. \\ 0.90874 & 0.800704 & 0.695991 & 0.650613 & 1. \\ 0.891533 & 0.695991 & 0.374921 & 0.181789 & 1. \\ 0.886197 & 0.650613 & 0.181789 & 0.00149764 & 1. \end{pmatrix}$$

In[10]:=
```
R = Array[(α#1 Z#1 + α#2 Z#2)/(α#1 + α#2) &, {8, 8}]; (* Distances matrix R_{p,q} *)
#[R] & /@ {SymmetricMatrixQ, ArrayPlot, MatrixForm}
```

Out[11]=

```
{False,
```



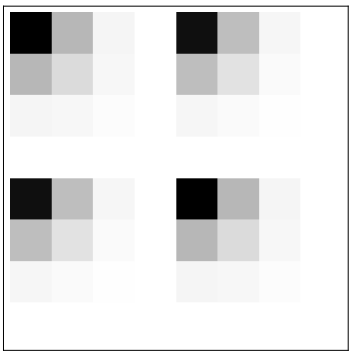
```
 ,
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
 (0.5)       (0.215276)  (0.0585161)  (0.009288
 (0.)        (0.)        (0.)         (0.
 (0.)        (0.)        (0.)         (0.
 (0.784724)  (0.5)       (0.184712)   (0.033044
 (0.)        (0.)        (0.)         (0.
 (0.)        (0.)        (0.)         (0.
 (0.941484)  (0.815288)  (0.5)        (0.13106
 (0.)        (0.)        (0.)         (0.
 (0.)        (0.)        (0.)         (0.
 (0.990712)  (0.966955)  (0.868931)   (0.5)
 (0.)        (0.)        (0.)         (0.)
 (0.)        (0.)        (0.)         (0.)
```

In[12]:=

```
S = Array[(π/(α#1 + α#2))^(3/2) K#1,#2 &, {8, 8}];(* Overlap matrix Sp,q *)
#[S] & /@ {SymmetricMatrixQ, ArrayPlot, MatrixForm}
```

Out[13]=

```
{True,
```

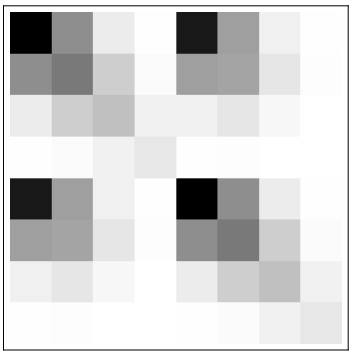


```
 ,
 46.2287    13.0602    1.85084    0.117043     43.494
 13.0602    6.64247    1.49148    0.112858     11.868
 1.85084    1.49148    0.716317   0.0961392    1.6500
 0.117043   0.112858   0.0961392  0.0419641    0.1037
 43.4941    11.8683    1.65009    0.103723     46.228
 11.8683    5.31865    1.03805    0.0734269    13.060
 1.65009    1.03805    0.268562   0.017477     1.8508
 0.103723   0.0734269  0.017477   0.000062847  0.1170
```

In[14]:=

```
T = Array[(α#1 α#2)/(α#1 + α#2) (3 - 2 (α#1 α#2)/(α#1 + α#2) Norm[Z#1 - Z#2]^2) S#1,#2 &, {8, 8}];
(* Kinetic energy matrix Tp,q *)
#[T] & /@ {SymmetricMatrixQ, ArrayPlot, MatrixForm}
```

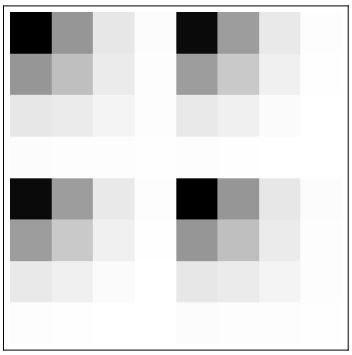
Out[15]=

{True,  ,

8.45632	3.74945	0.637503	0.042422	7.6
3.74945	4.42916	1.62162	0.145532	3.
0.637503	1.62162	2.1082	0.491726	0.5
0.042422	0.145532	0.491726	0.818786	0.03
7.63269	3.1899	0.524852	0.0345662	8.4
3.1899	3.02094	0.85594	0.0675523	3.7
0.524852	0.85594	0.273461	-0.0122114	0.6
0.0345662	0.0675523	-0.0122114	-0.00409065	0.0

In[16]:=
```
V = Total@Table[Array[If[Norm[R#1,#2 - U] == 0, -(2 π)/(α#1 + α#2) K#1,#2,
    -S#1,#2 1/Norm[R#1,#2 - U] Erf[Sqrt[α#1 + α#2] Norm[R#1,#2 - U]]] &, {8, 8}], {U, P}];
 (* Potential energy matrix Vp,q *)
#[V] & /@ {SymmetricMatrixQ, ArrayPlot, MatrixForm}
```

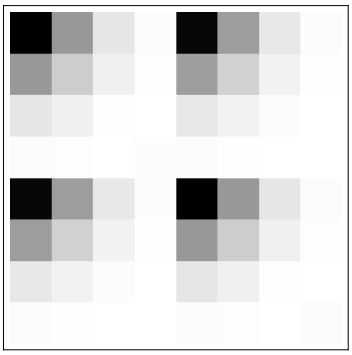
Out[17]=

{True,  ,

-49.5733	-20.4017	-4.78952	-0.595591	-47
-20.4017	-12.4983	-4.06016	-0.579931	-19
-4.78952	-4.06016	-2.31383	-0.515863	-4.
-0.595591	-0.579931	-0.515863	-0.283481	-0.
-47.5078	-19.0125	-4.33849	-0.528623	-49
-19.0125	-10.5321	-2.948	-0.378339	-20
-4.33849	-2.948	-0.900982	-0.0903486	-4.
-0.528623	-0.378339	-0.0903486	-0.00025131	-0.

In[18]:=
```
H = T + V;
#[H] & /@ {SymmetricMatrixQ, ArrayPlot, MatrixForm}
```

Out[19]=

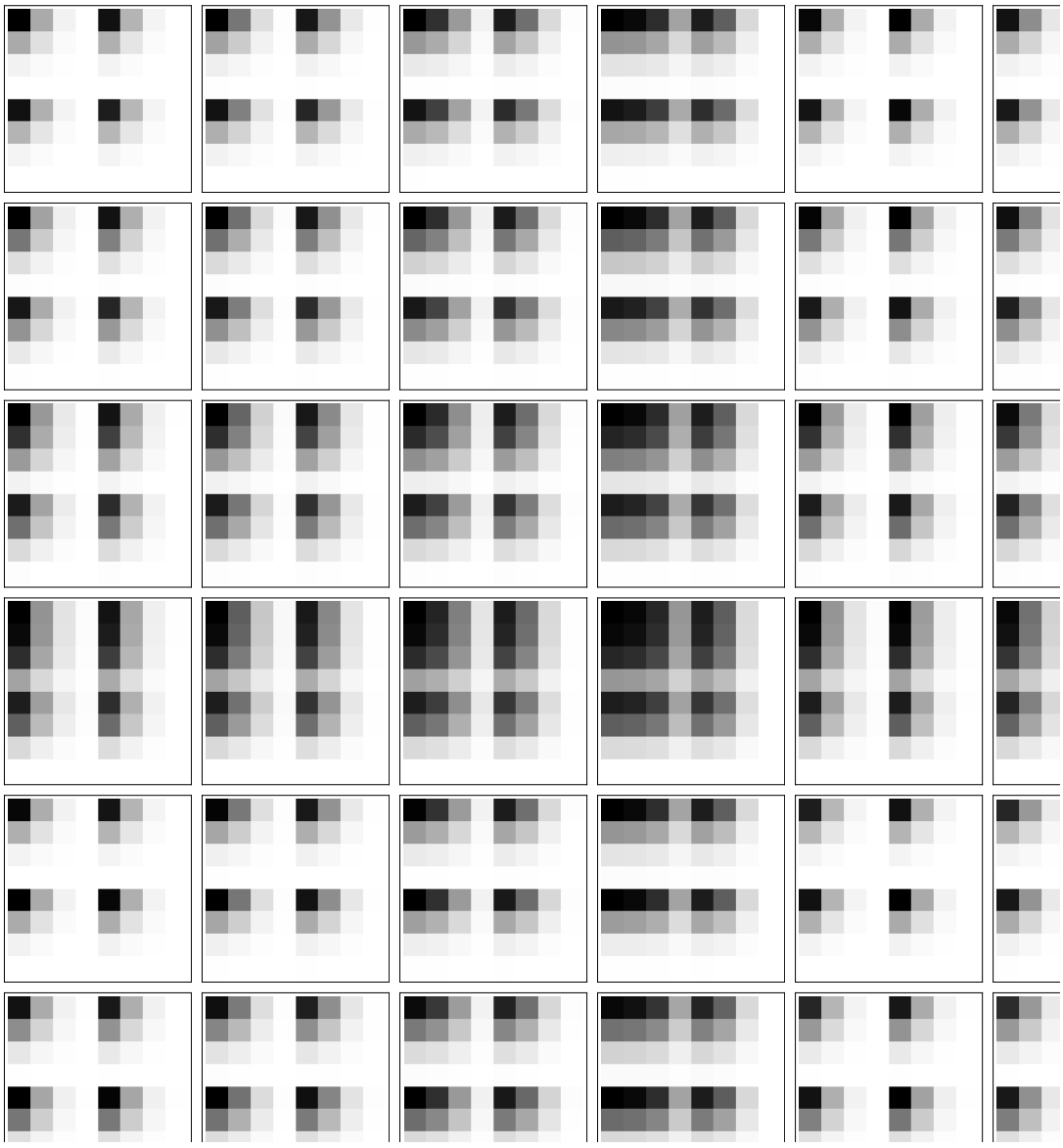
{True,  ,

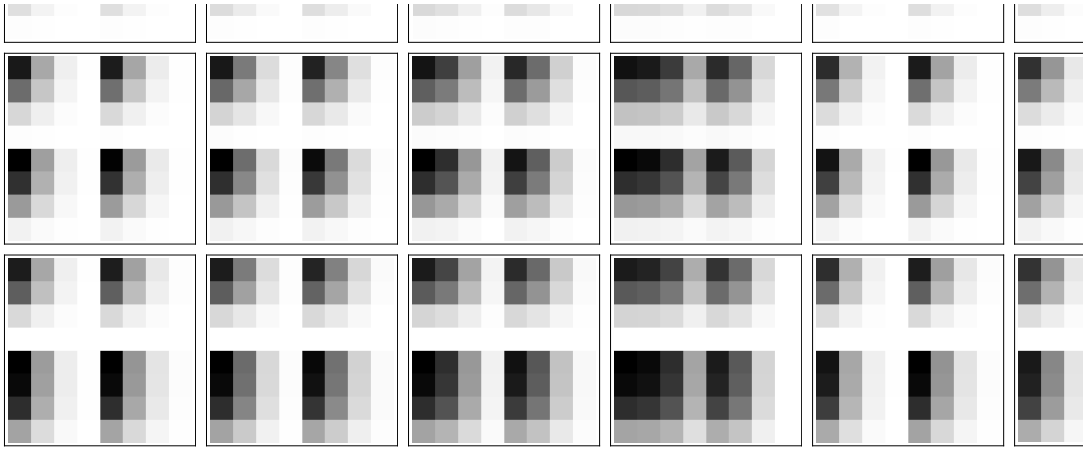
-41.117	-16.6522	-4.15202	-0.553169	-39
-16.6522	-8.06909	-2.43854	-0.434398	-15
-4.15202	-2.43854	-0.205623	-0.0241367	-3.
-0.553169	-0.434398	-0.0241367	0.535304	-0.
-39.8751	-15.8226	-3.81364	-0.494057	-4
-15.8226	-7.51116	-2.09206	-0.310786	-16
-3.81364	-2.09206	-0.627521	-0.10256	-4.
-0.494057	-0.310786	-0.10256	-0.00434196	-0.

In[20]:=
```
Q = Array[S_{#1,#3} S_{#2,#4} If[Norm[R_{#1,#3} - R_{#2,#4}] == 0, 2/Sqrt[π] Sqrt[((α_{#1} + α_{#3}) (α_{#2} + α_{#4}))/(α_{#1} + α_{#3} + α_{#2} + α_{#4})],
    1/Norm[R_{#1,#3} - R_{#2,#4}] Erf[Sqrt[((α_{#1} + α_{#3}) (α_{#2} + α_{#4}))/(α_{#1} + α_{#3} + α_{#2} + α_{#4})] Norm[R_{#1,#3} - R_{#2,#4}]]] &,
   {8, 8, 8, 8}]; (* Q_{p,r,q,s} matrix, it's an 8 by 8
 matrix containing 8 by 8 matrices *)
{SymmetricMatrixQ[Q],
 Grid@Array[ArrayPlot[Q_{#1,#2}, ImageSize → Tiny] &, {8, 8}], MatrixForm@Q}
```

Out[21]=

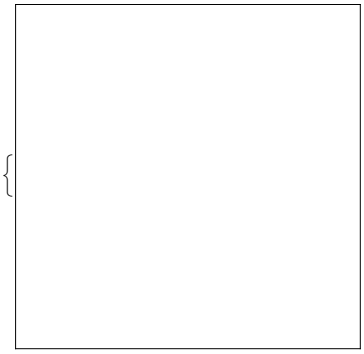
{False,





In[22]:=
```
P = ConstantArray[0., {8, 8}];
#[P] & /@ {ArrayPlot, MatrixForm}
```

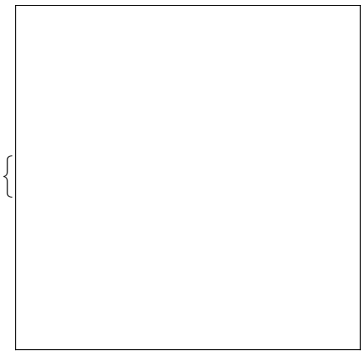
Out[23]=



$$\left\{ \quad , \begin{pmatrix} 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix} \right\}$$

In[24]:=
```
G = Array[Sum[Sum[P_{r,s} (2 Q_{#1,r,#2,s} - Q_{#1,r,s,#2}), {s, 1, 8}], {r, 1, 8}] &, {8, 8}] (* G_{p,q} matrix *);
#[G] & /@ {ArrayPlot, MatrixForm}
```

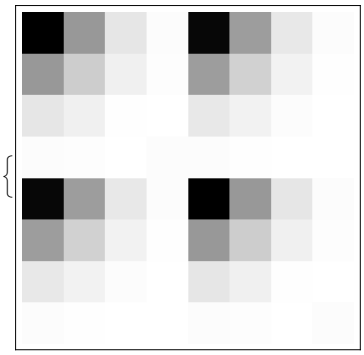
Out[25]=



$$\left\{ \quad , \begin{pmatrix} 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix} \right\}$$

In[26]:=
```
F = H + 1/2 G;
#[F] & /@ {ArrayPlot, MatrixForm}
```

Out[27]=

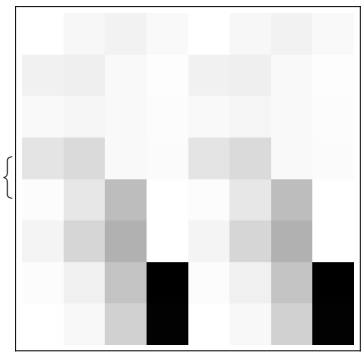


$$\begin{pmatrix} -41.117 & -16.6522 & -4.15202 & -0.553169 & -39.8751 \\ -16.6522 & -8.06909 & -2.43854 & -0.434398 & -15.8226 \\ -4.15202 & -2.43854 & -0.205623 & -0.0241367 & -3.81364 \\ -0.553169 & -0.434398 & -0.0241367 & 0.535304 & -0.494057 \\ -39.8751 & -15.8226 & -3.81364 & -0.494057 & -41.117 \\ -15.8226 & -7.51116 & -2.09206 & -0.310786 & -16.6522 \\ -3.81364 & -2.09206 & -0.627521 & -0.10256 & -4.15202 \\ -0.494057 & -0.310786 & -0.10256 & -0.00434196 & -0.553169 \end{pmatrix}$$

In[28]:=
```
MatrixNormalize[v_, M_] := v/Sqrt[v.M.v];
```

In[29]:=
```
(* Doing the first eigensystem calculation as a testing ground *)
{λ, ψ} = Eigensystem[{F, S}];(* Calculate the eigenvalues and eigenvectors *)
{λ, ψ} = {λ[[#]], ψ[[#]]} &@Ordering[λ]; (* Sort them in ascending order *)
ψ = MatrixNormalize[#, S] & /@ ψ;
(* Normalize them w.r.t. the overlap matrix S *)
If[AllTrue[Thread[#.S.# ≠ 1] & /@ ψ, TrueQ], Abort[], Nothing];
(* Perform a numerical divergence check *)
#[ψ] & /@ {ArrayPlot, MatrixForm}
```

Out[33]=



$$\begin{pmatrix} -0.0053109 & -0.138633 & -0.227877 & -0.103876 & -0.0053109 \\ -0.247679 & -0.290428 & -0.112811 & -0.0503944 & 0.247679 \\ 0.108453 & -0.18563 & -0.114518 & -0.0553259 & 0.108453 \\ -0.480872 & 0.64799 & 0.112376 & 0.07753 & 0.480872 \\ -0.0670001 & 0.421322 & -1.13493 & 0.00410123 & -0.0670001 \\ 0.195647 & -0.715908 & 1.35432 & 0.0142581 & -0.195647 \\ 0.0648705 & -0.272204 & 1.03771 & -4.39883 & -0.0648705 \\ -0.0163136 & 0.129456 & -0.793669 & 4.35218 & -0.0163136 \end{pmatrix}$$

In[34]:=
```
ϵ = 10^-6 // N; (* Accuracy level *)
```

In[35]:=
```
(* Reset before the while-loop to assert correctness *)
λ = ConstantArray[0., {8}];
ξ = λ + ϵ;
```

In[37]:=
```
Δ = {}; (* List of the norm of the differences *)
```

In[38]:=
```
(* While the new and previous eigenvalues are still different enough... *)
While[Norm[ξ - λ] > ϵ,
   AppendTo[Δ, Norm[ξ - λ]];
   (* Append the norm of the difference to the Δ list *)
   ξ = λ; (* Overwrite the previous eigenvalues *)
   G = Array[Sum[Sum[P_{r,s} (2 Q_{#1,r,#2,s} - Q_{#1,r,s,#2}) &, {8, 8}], {r, 1, 8}], {s, 1, 8}] (* G_{p,q} matrix *);
   F = H + 1/2 G; (* Compute the Fock operator from the G matrix *)
   {λ, ψ} = Eigensystem[{F, S}];
   (* Calculate the eigenvalues and eigenvectors *)
   {λ, ψ} = {λ[[#]], ψ[[#]]} &@Ordering[λ]; (* Sort them in ascending order *)
   ψ = MatrixNormalize[#, S] & /@ ψ;
   (* Normalize them w.r.t. the overlap matrix S *)
   If[AllTrue[Thread[#.S.# ≠ 1] & /@ ψ, TrueQ], Abort[], Nothing];
   (* Perform a numerical divergence check *)
   (* Calculate the density matrix from the
    eigenvector ψ_1 associated with the minimal eigenvalue *)
   P = 2 TensorProduct[ψ_1, ψ_1];
  ] // AbsoluteTiming // First
```

Out[38]=

0.052874

In[39]:=
```
(* Return the first eigenvalue and the equilibrium bonding energy,
where 1/d is the E_nucl term *)
Quantity[#, "HartreeEnergy"] & /@ {λ_1, Total[P (H + 1/4 G), 2] + 1/d}
UnitConvert[#, "Electronvolts"] & /@ %
```

Out[39]=

$\{-0.669956\, E_h,\ -1.07855\, E_h\}$

Out[40]=

$\{-18.2304\ \text{eV},\ -29.3488\ \text{eV}\}$

In[41]:=
```
ListStepPlot[Drop[Δ, 1], PlotRange → Full]
(* Measurements give 0.05 seconds for the convergence time for an SCF loop *)
```

Out[41]=