Solving Schrödinger equation numerically

Basic idea on working out the energy eigenvalues numerically is very simple. Just solve the Schrödinger equation with a guessed energy, and it always makes the wave function blow up at the infinity. Do trial-and-error to find an energy for which the wave function is tamed up to a very large value of the radius. In this notebook, I'll give a few examples so that you get an idea how to do it. You are welcome to use my commands modified for your purpose.

I restrict the discussions to spherically symmetric systems.

The Schrödinger equation for the radial wave function is
$\left(\frac{-\hbar^2}{2m}\frac{1}{r}\frac{d^2}{dr^2}r+\frac{\hbar^2}{2m}\frac{l(l+1)}{r^2}+V(r)\right)R(r)=E\,R(r)$.
Multipying $2m/\hbar^2$ on the both sides,
$\left(-\frac{1}{r}\frac{d^2}{dr^2}r+\frac{l(l+1)}{r^2}+U(r)\right)R(r)=\varepsilon\,R(r)$.
Here, I introduced the notation $U(r)=\frac{2m}{\hbar^2}V(r)$, $\varepsilon=\frac{2m}{\hbar^2}E$

Finally regular solutions at the origin are of the form $R(r)=r^l\,\rho(r)$. Note that
$\frac{1}{r}\frac{d^2}{dr^2}r\,R(r)=\frac{1}{r}\frac{d^2}{dr^2}r^{l+1}\,\rho(r)=\frac{1}{r}\left((l+1)\,l\,r^{l-1}\,\rho+2\,(l+1)\,r^l\,\rho'+r^{l+1}\,\rho''\right)$
The first term cancels the centrifugal potential term, and we find
$-\rho''(r)-\frac{2(l+1)}{r}\rho'(r)+U(r)\,\rho(r)=\varepsilon\,\rho(r)$

If the potential is regular at the origin, the singular term $\frac{2(l+1)}{r}$needs to be cancelled by $\rho'(0)=0$. If the potential is not regular, such as the Coulomb potential, an additional care is needed as explained below.

3D harmonic oscillator

We know the eigenvalues of this problem from 221A.

$E=\frac{3}{2}\hbar\,\omega$ $(l=0)$
$E=\frac{5}{2}\hbar\,\omega$ $(l=1)$
$E=\frac{7}{2}\hbar\,\omega$ $(l=0,2)$
$E=\frac{9}{2}\hbar\,\omega$ $(l=1,3)$
$E=\frac{11}{2}\hbar\,\omega$ $(l=0,2,4)$
$E=\frac{13}{2}\hbar\,\omega$ $(l=1,3,5)$
etc.

For numerical calculations, I set $\hbar=m=\omega=1$. Then $U(r)=r^2$, $\varepsilon=2E$.

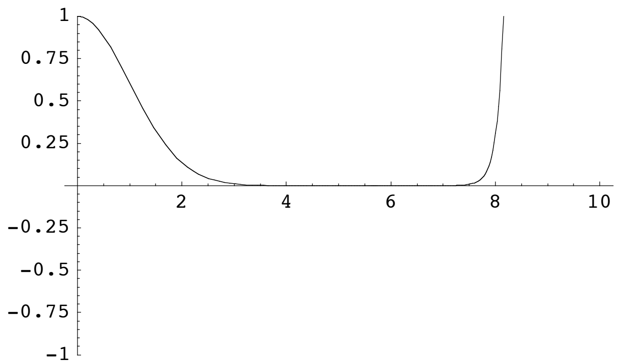
l=0

---

no nodes

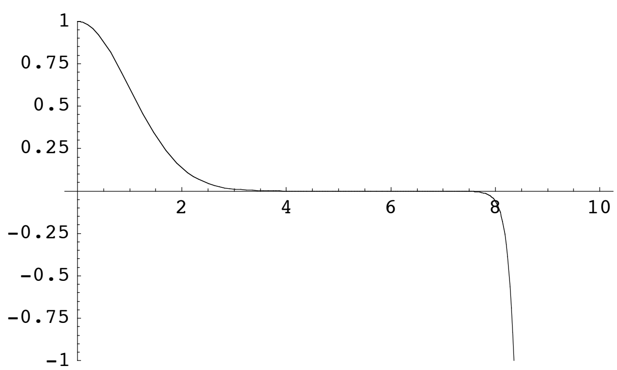
With no nodes (ground state), we expect $\varepsilon=3$.

```
In[83]:= sol = NDSolve[{-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r² ρ[r] == ε ρ[r] /. {l → 0} /.
             {ε → 2.99999997876}, ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
         Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```




```
Out[83]= - Graphics -
```

```
In[84]:= sol = NDSolve[{-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r² ρ[r] == ε ρ[r] /. {l → 0} /.
             {ε → 2.99999997877}, ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
         Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```




```
Out[84]= - Graphics -
```

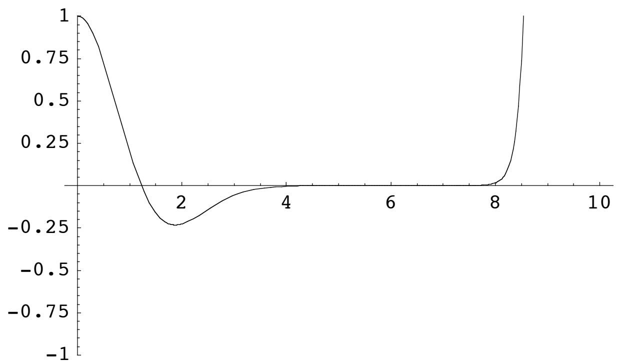
I find the eigenvalue to be $2.99999997876 < \varepsilon < 2.99999997877$, compared to the exact result $\varepsilon = 3$. You get a sense on the numerical accuracy, at well as the eigenvalue itself.

one node

Now for the higher levels. With one node, we expect $\varepsilon = 7$,

In[111]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 0} /. {ε → 6.9999999265},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

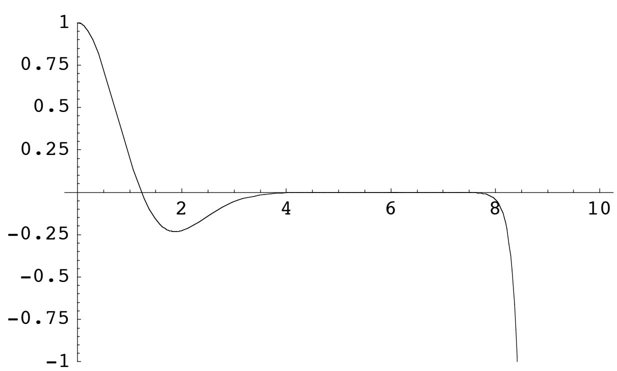


Out[111]=

- Graphics -

In[112]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 0} /. {ε → 6.9999999264},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[112]=

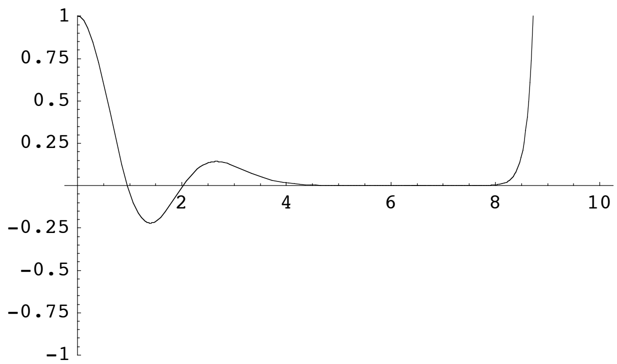
- Graphics -

two nodes

With two nodes, we expect $\varepsilon = 11$,

In[143]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 0} /. {ε → 11.000000071},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

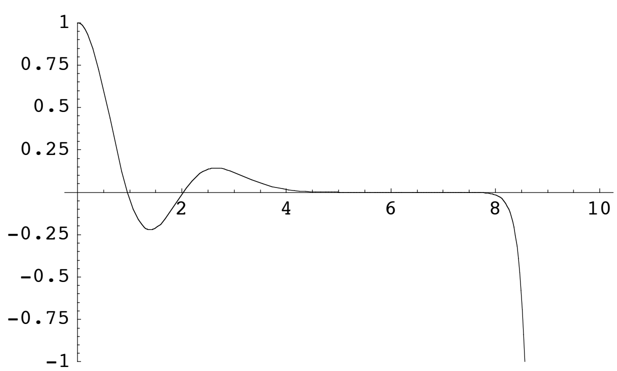


Out[143]=

- Graphics -

In[144]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 0} /. {ε → 11.000000072},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[144]=

- Graphics -

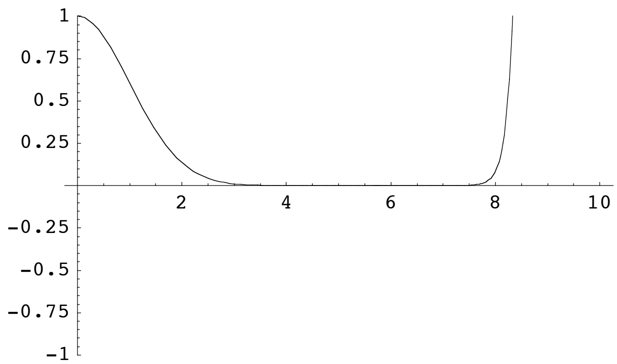
l=1

no nodes

With no nodes (ground state), we expect $\varepsilon = 5$.

In[162]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 1} /. {ε → 4.9999999971},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

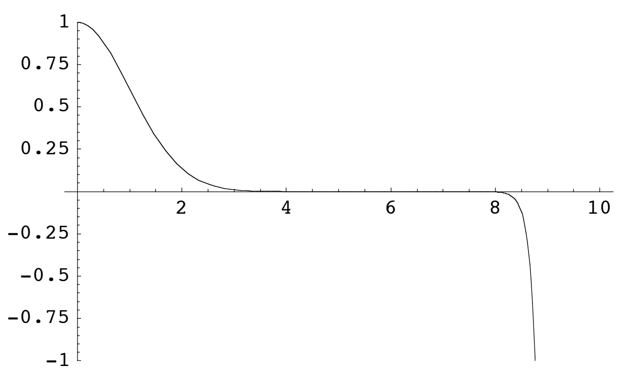


Out[162]=

- Graphics -

In[160]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 1} /. {ε → 4.9999999972},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[160]=

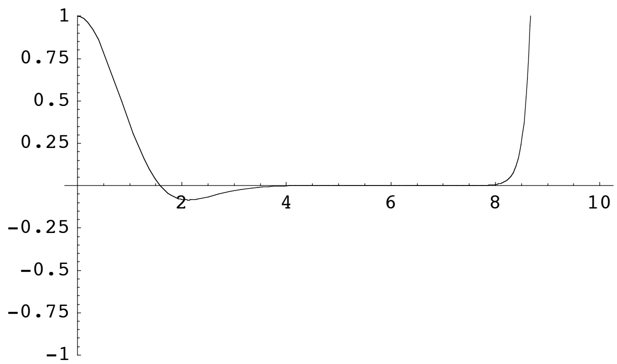
- Graphics -

one node

Now for the higher levels. With one node, we expect $\varepsilon = 9$,

In[193]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 1} /. {ε → 9.00000003},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

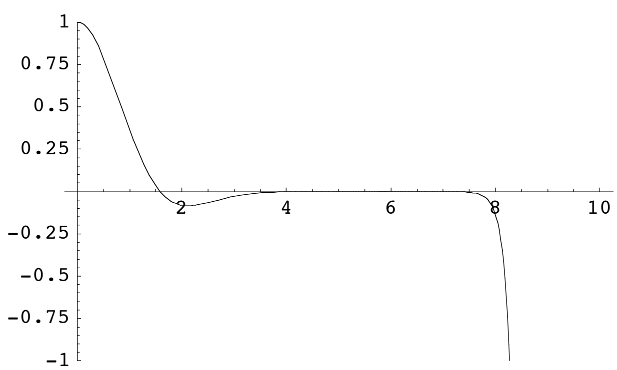


Out[193]=

- Graphics -

In[194]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 1} /. {ε → 9.00000002},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[194]=

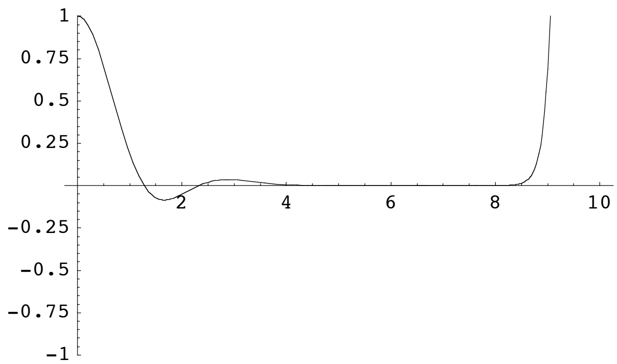
- Graphics -

two nodes

With two nodes, we expect $\varepsilon = 11$,

In[212]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 1} /. {ε → 13.000000115},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

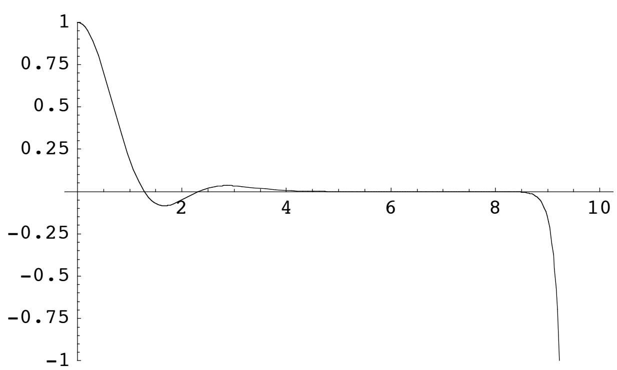


Out[212]=

- Graphics -

In[213]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 1} /. {ε → 13.000000116},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[213]=

- Graphics -

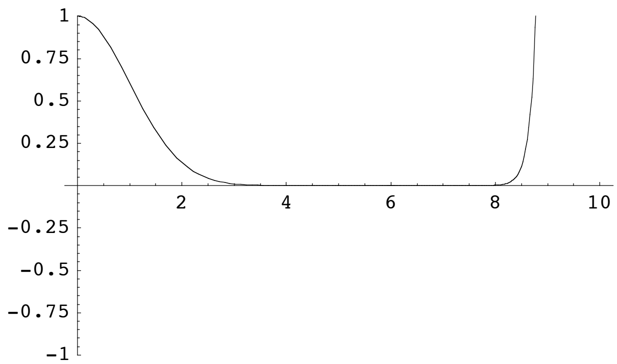
l=2

no nodes

With no nodes (ground state), we expect $\varepsilon = 7$.

In[233]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 2} /. {ε → 6.9999997058},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

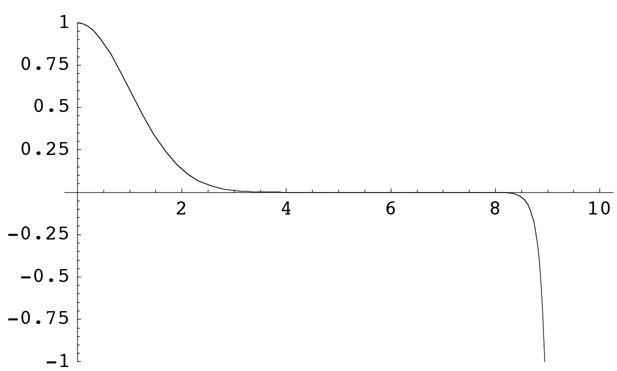


Out[233]=

- Graphics -

In[234]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 2} /. {ε → 6.9999997059},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[234]=

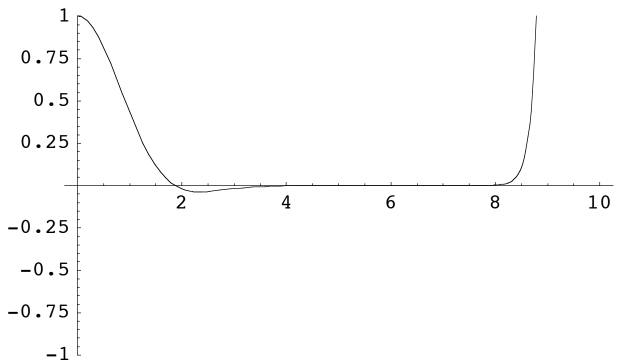
- Graphics -

one node

Now for the higher levels. With one node, we expect $\varepsilon = 11$,

In[252]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 2} /. {ε → 10.99999983},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

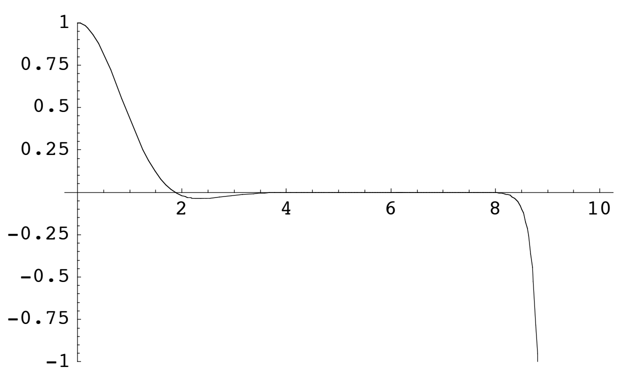


Out[252]=

- Graphics -

In[253]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 2} /. {ε → 10.99999982},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[253]=

- Graphics -

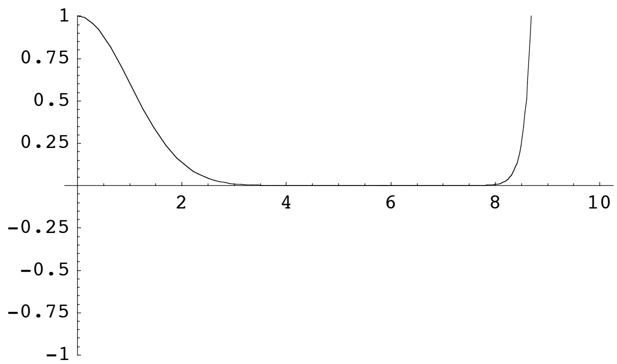
l=3

no nodes

With no nodes (ground state), we expect $\varepsilon = 9$.

In[265]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 3} /. {ε → 8.99999966},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

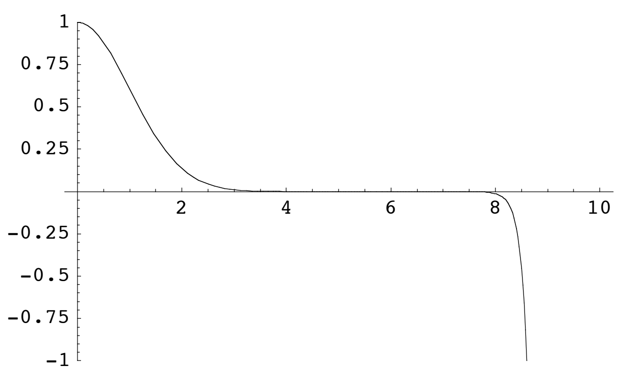


Out[265]=

```
- Graphics -
```

In[266]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 3} /. {ε → 8.99999967},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[266]=

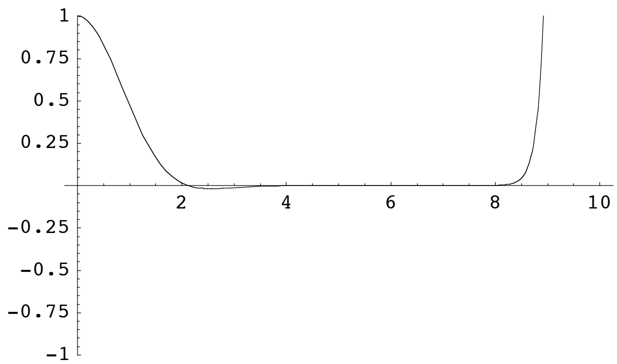
```
- Graphics -
```

one node

Now for the higher levels. With one node, we expect $\varepsilon = 13$,

In[279]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 3} /. {ε → 12.9999953},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

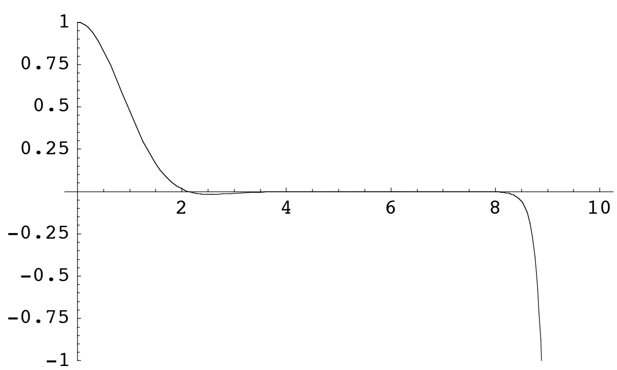


Out[279]=

- Graphics -

In[280]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, 2 (l + 1)/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 3} /. {ε → 12.9999952},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[280]=

- Graphics -

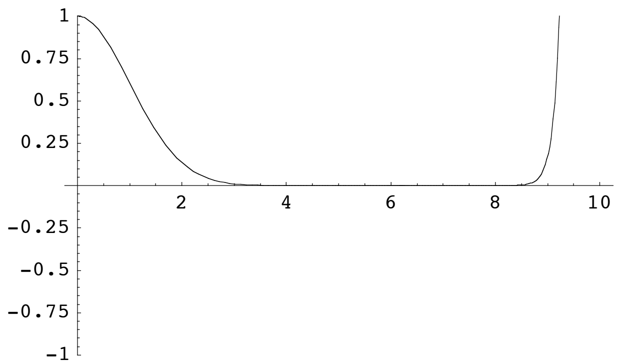
l=4

no nodes

With no nodes (ground state), we expect $\varepsilon = 11$.

In[297]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 4} /. {ε → 10.999999161},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

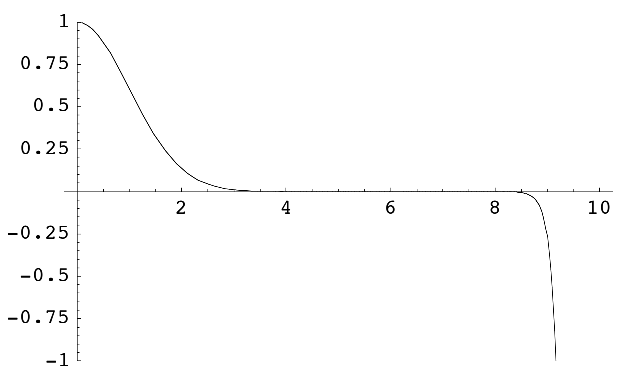


Out[297]=

```
- Graphics -
```

In[298]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 4} /. {ε → 10.999999162},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[298]=

```
- Graphics -
```

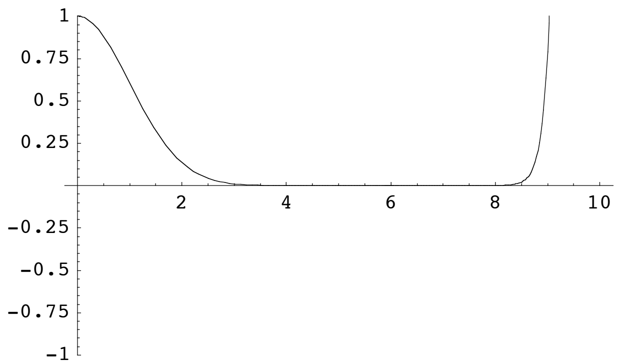
l=5

no nodes

With no nodes (ground state), we expect $\varepsilon = 13$.

In[308]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 5} /. {ε → 13.0000004},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```

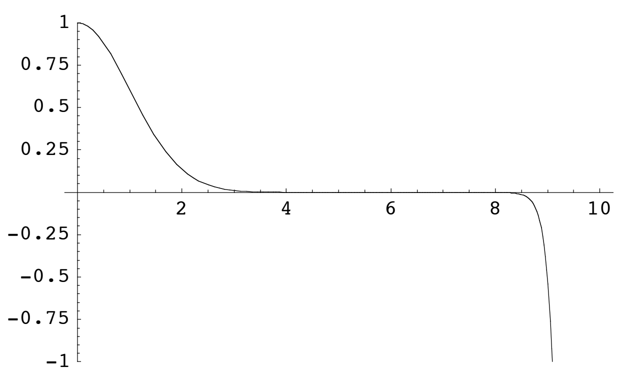


Out[308]=

- Graphics -

In[309]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r] ρ'[r] + r^2 ρ[r] == ε ρ[r] /. {l → 5} /. {ε → 13.0000005},
   ρ[0] == 1, ρ'[0] == 0}, ρ, {r, 0, 10}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 10}, PlotRange → {-1, 1}]
```



Out[309]=

- Graphics -

Coulomb potential (hydrogen atom)

The potential is $V(r) = -\frac{e^2}{r}$, $U(r) = -\frac{2m}{\hbar^2}\frac{e^2}{r} = -2\frac{a_B}{r}$. The Schrödinger equation is $-\rho''(r) - \frac{2(l+1)}{r}\rho'(r) - 2\frac{a_B}{r}\rho(r) = \varepsilon\,\rho(r)$.
Close to the origin, we expand $\rho(r) = \rho(0) + \rho'(0)\,r + O(r^2)$. The singiularity in $-\frac{2(l+1)}{r}\rho'(r)$ must be canceled by the corresponding singularity due to the Coulomb potential $-2\frac{a_B}{r}\rho(0)$, and hence $2(l+1)\rho'(0) + 2a_B\,\rho(0) = 0$, $\rho'(0) = -\frac{a_B}{l+1}\rho(0)$. Because we don't care the overall normalization of the wave function, we can just set $\rho(0) = 1$, $\rho'(0) = -a_B/(l+1)$.

For numerical calculations, we use the atomic unit, $e^2 = m = \hbar = 1$. Then the energy levels are $E = -\frac{e^4 m}{2\hbar^2 n^2} = -(2n^2)^{-1}$, and $\varepsilon = -n^{-2}$.

l=0

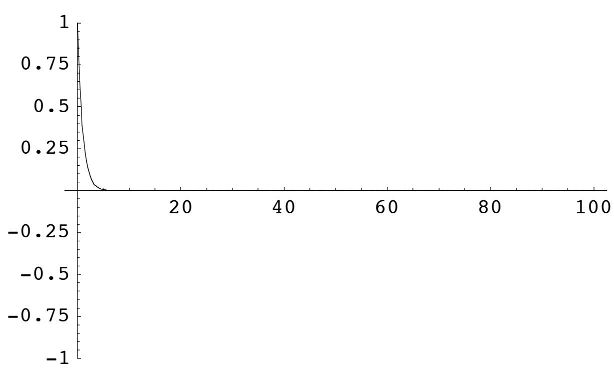
---

no nodes

With no nodes (ground state), we expect $\varepsilon = -1$.

In[367]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 0} /. {ε → -1}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[367]=

- Graphics -

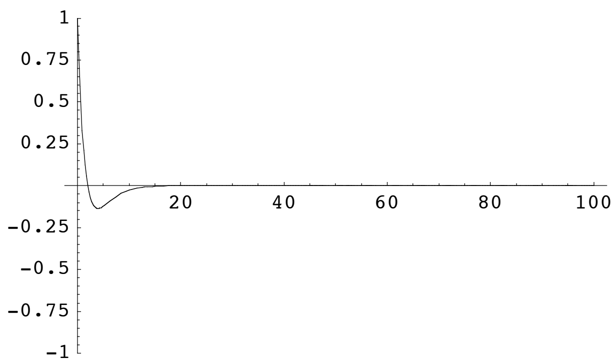
---

one node

Now for the higher levels. With one node, we expect $\varepsilon = -0.25$,

In[424]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 0} /. {ε → -0.250000003824023}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[424]=

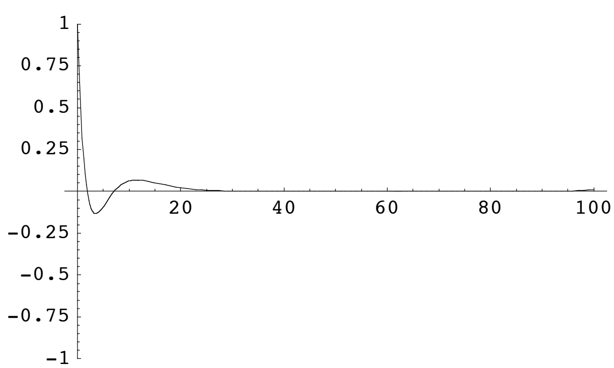
- Graphics -

two nodes

With two nodes, we expect $\varepsilon = \frac{-1}{9}$,

In[415]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 0} /. {ε → -0.1111111209}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[415]=

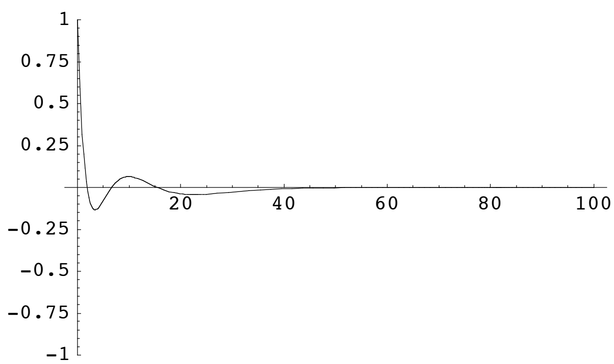
- Graphics -

three nodes

With two nodes, we expect $\varepsilon = \frac{-1}{16}$,

In[433]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 0} /. {ε → -0.0625}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[433]=

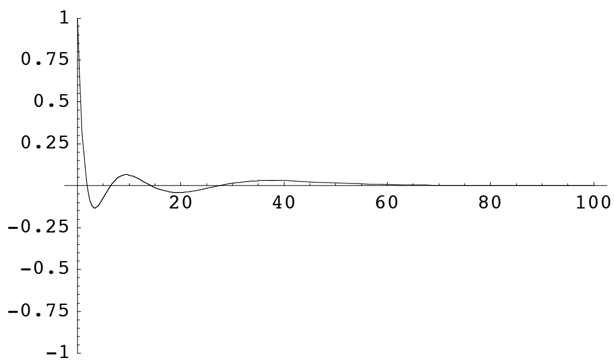
```
- Graphics -
```

four nodes

With two nodes, we expect $\varepsilon = \frac{-1}{25}$,

In[434]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 0} /. {ε → -0.04}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[434]=

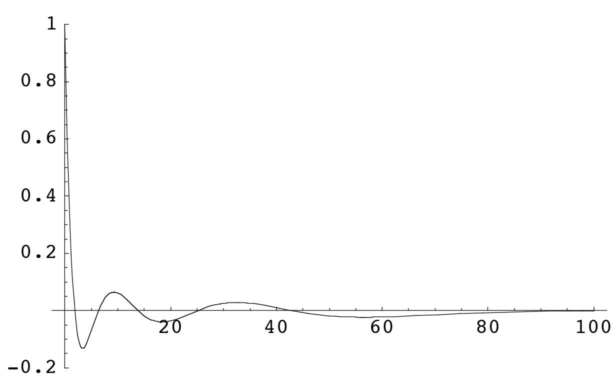
- Graphics -

five nodes

With two nodes, we expect $\varepsilon = \frac{-1}{36}$,

In[450]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 0} /. {ε → -0.0277777}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-0.2, 1}]
```



Out[450]=

- Graphics -

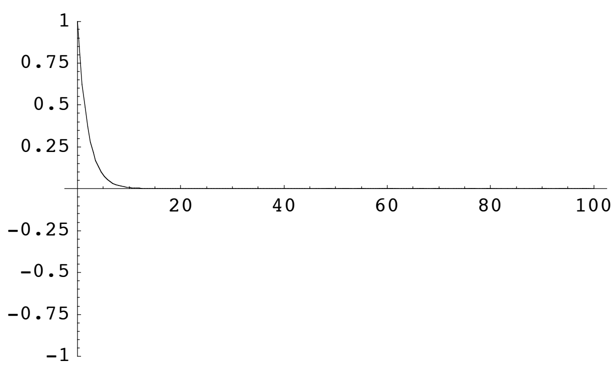
l=1

no nodes

With no nodes (ground state), we expect $\varepsilon = \frac{-1}{4}$.

In[436]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 1} /. {ε → -0.25}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[436]=

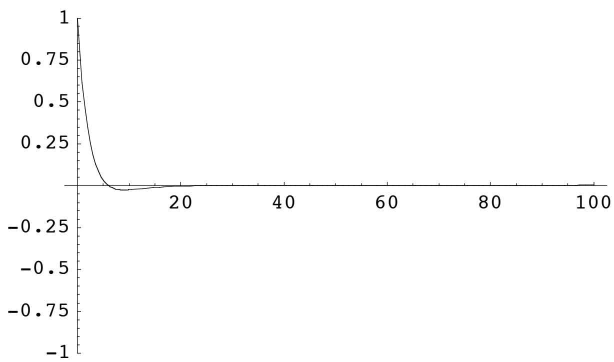
```
- Graphics -
```

one node

Now for the higher levels. With one node, we expect $\varepsilon = \frac{-1}{9}$,

In[444]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 1} /. {ε → -0.111111098}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[444]=

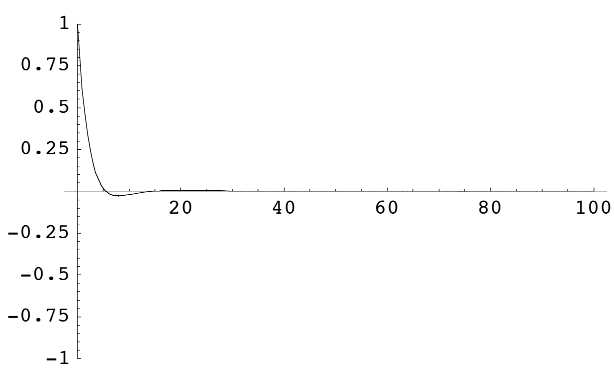
- Graphics -

two nodes

With two nodes, we expect $\varepsilon = \frac{-1}{16}$,

In[445]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 1} /. {ε → -0.0625}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-1, 1}]
```



Out[445]=

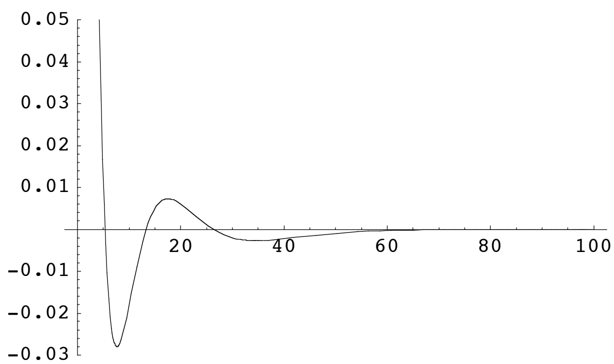
- Graphics -

three nodes

With two nodes, we expect $\varepsilon = \frac{-1}{25}$,

In[456]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 1} /. {ε → -0.04}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-0.03, 0.05}]
```



Out[456]=

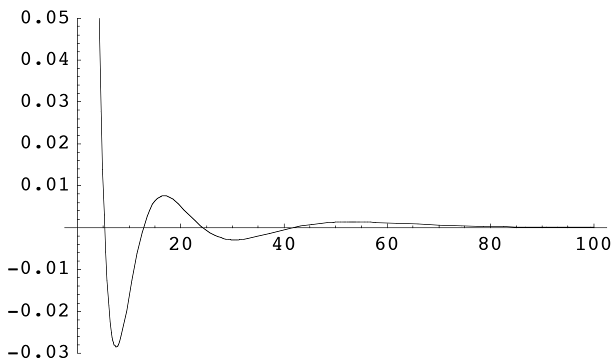
```
- Graphics -
```

four nodes

With two nodes, we expect $\varepsilon = \frac{-1}{36}$,

In[460]:=

```
sol = NDSolve[
  {-ρ''[r] - If[r == 0, 0, (2 (l + 1))/r ρ'[r] + 2/r ρ[r]] == ε ρ[r], ρ[0] == 1, ρ'[0] == -1/(l + 1)} /.
    {l → 1} /. {ε → -0.0277777}, ρ, {r, 0, 100}];
Plot[Evaluate[ρ[r] /. sol], {r, 0, 100}, PlotRange → {-0.03, 0.05}]
```



Out[460]=

- Graphics -