Example of a Floquet matrix calculation for angular momentum

The Hamiltonian is
H=hbar .L_z+hbar.e .Cos[w.t]L_x
The fixed precession frequency around the z-axis is taken as unity. The driving frequency is w and the driving strength is e.
We will assume that the total angular momentum eigenvalue is l=1, so all matrices are only 3 x 3.

Below, the prob. amplitudes for Lz measurements to give (+1,0,-1) are respectively (ap(t), a0(t) , am(t)).
We first initialise the driving strength and the frequency, and consider an initial eigenstate Lz=+1, so that (ap(0)=1, a0(0)=0, am(0)=0)

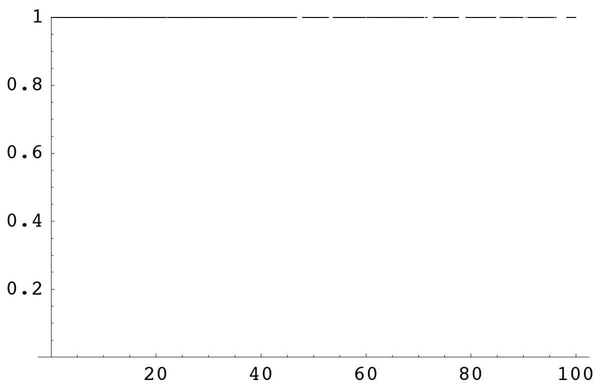
```
{e = 0.1, w = 1.0};

NDSolve[{ap'[t] == -I * ap[t] - I * (e / Sqrt[2]) * Cos[w * t] * a0[t],
          a0'[t] == -I * (e / Sqrt[2]) * Cos[w * t] * (ap[t] + am[t]),
         am'[t] == I * am[t] - I * (e / Sqrt[2]) * Cos[w * t] * a0[t],
          ap[0] == 1.0, a0[0] == 0.0, am[0] == 0.0}, {ap, a0, am}, {t, 0.0, 100.0}]
```

```
{{ap → InterpolatingFunction[{{0., 100.}}, <>],
  a0 → InterpolatingFunction[{{0., 100.}}, <>],
  am → InterpolatingFunction[{{0., 100.}}, <>]}}
```

First check that state remains normalised for all time by adding the probabilities |ap(t)|^2 + |a0(t)|^2 + |am(t)|^2 and check to see that it is untiy for all time.

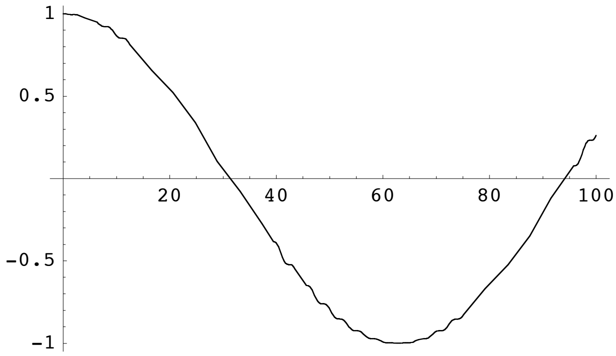
```
Plot[(Abs[ap[x]]^2 + Abs[a0[x]]^2 + Abs[am[x]]^2) /. %,
 {x, 0.0, 100.0}, PlotRange → {0.0, 1.0}]
```




```
- Graphics -
```

Next, compute average value of Lz(t): <Lz(t)>= (+1)|ap(t)|^2 + (-1)|am(t)|^2.

```
Plot[(Abs[ap[x]]^2 - Abs[am[x]]^2) /. %, {x, 0.0, 100.0}, PlotRange → All]
```




```
- Graphics -
```

This oscillation is what we expect. The system is driven into a periodic oscialltion so that the angular momentum vector is rotating around the Lx axis. This is called a"Rabi oscillation" .

We now compute the Floquet matrix. First top row, then second row then last row.

```
In[9]:=  {e = 0.1, w = 1.0};

In[13]:= NDSolve[{ap'[t] == -I * ap[t] - I * (e / Sqrt[2]) * Cos[w * t] * a0[t],
                  a0'[t] == -I * (e / Sqrt[2]) * Cos[w * t] * (ap[t] + am[t]),
                 am'[t] == I * am[t] - I * (e / Sqrt[2]) * Cos[w * t] * a0[t],
                  ap[0] == 1.0, a0[0] == 0.0, am[0] == 0.0}, {ap, a0, am}, {t, 0.0, 100.0}]

Out[13]= {{ap → InterpolatingFunction[{{0., 100.}}, <>],
           a0 → InterpolatingFunction[{{0., 100.}}, <>],
           am → InterpolatingFunction[{{0., 100.}}, <>]}}

In[14]:= top = {ap[2 * Pi], a0[2 * Pi], am[2 * Pi]} /. %

Out[14]= {{0.975528 + 0.00386514 i, 0.000432766 - 0.218474 i, -0.0244641 - 1.0882 × 10^-9 i}}
```

Now we find the middle row.

```
In[15]:= NDSolve[{ap'[t] == -I * ap[t] - I * (e / Sqrt[2]) * Cos[w * t] * a0[t],
                  a0'[t] == -I * (e / Sqrt[2]) * Cos[w * t] * (ap[t] + am[t]),
                 am'[t] == I * am[t] - I * (e / Sqrt[2]) * Cos[w * t] * a0[t],
                  ap[0] == 0.0, a0[0] == 1.0, am[0] == 0.0}, {ap, a0, am}, {t, 0.0, 100.0}]

Out[15]= {{ap → InterpolatingFunction[{{0., 100.}}, <>],
           a0 → InterpolatingFunction[{{0., 100.}}, <>],
           am → InterpolatingFunction[{{0., 100.}}, <>]}}

In[16]:= mid = {ap[2 * Pi], a0[2 * Pi], am[2 * Pi]} /. %

Out[16]= {{0.000432724 - 0.218474 i, 0.951072 + 0. i, -0.000432724 - 0.218474 i}}
```

And finally the bottom row,

```
In[17]:= NDSolve[{ap'[t] == -I*ap[t] - I*(e/Sqrt[2])*Cos[w*t]*a0[t],
                 a0'[t] == -I*(e/Sqrt[2])*Cos[w*t]*(ap[t] + am[t]),
                 am'[t] == I*am[t] - I*(e/Sqrt[2])*Cos[w*t]*a0[t],
                 ap[0] == 0.0, a0[0] == 0.0, am[0] == 1.0}, {ap, a0, am}, {t, 0.0, 100.0}]

Out[17]= {{ap → InterpolatingFunction[{{0., 100.}}, <>],
           a0 → InterpolatingFunction[{{0., 100.}}, <>],
           am → InterpolatingFunction[{{0., 100.}}, <>]}}

In[18]:= bot = {ap[2*Pi], a0[2*Pi], am[2*Pi]} /. %

Out[18]= {{-0.0244641 + 1.0882×10^-9 i, -0.000432766 - 0.218474 i, 0.975528 - 0.00386514 i}}

In[19]:= F = {top[[1]], mid[[1]], bot[[1]]}

Out[19]= {{0.975528 + 0.00386514 i, 0.000432766 - 0.218474 i, -0.0244641 - 1.0882×10^-9 i},
          {0.000432724 - 0.218474 i, 0.951072 + 0. i, -0.000432724 - 0.218474 i},
          {-0.0244641 + 1.0882×10^-9 i, -0.000432766 - 0.218474 i, 0.975528 - 0.00386514 i}}
```

Compute eigenvalues of F:

```
In[20]:= Eigenvalues[F]

Out[20]= {1. - 1.03216×10^-16 i, 0.951064 - 0.308993 i, 0.951064 + 0.308993 i}
```

Check that they all lie on the unit circle.

```
In[21]:= Abs[%]^2

Out[21]= {1., 1., 1.}
```

We can also find the eigenvectors (remember we are working in the Lz basis)

```
In[22]:= Eigenvectors[F]

Out[22]= {{-0.707051 - 5.18475×10^-7 i, -0.0125088 - 4.58628×10^-9 i, 0.707051 + 0. i},
          {0.493746 - 1.68137×10^-7 i, 0.707051 + 0. i, 0.506255 + 1.43245×10^-8 i},
          {-0.506255 + 1.43245×10^-8 i, 0.707051 + 0. i, -0.493746 - 1.68137×10^-7 i}}
```

The very small imaginary components here can be approxiated as zero so that all eigenvectors are real.