.. code:: ipython3

```
import matplotlib.pyplot as plt
import matplotlib as m
import numpy as np
from qutip import *
from scipy import *
from tqdm import tqdm
import os
import time
from scipy.optimize import leastsq
import math
import pandas as pd
import warnings
from io import StringIO

warnings.simplefilter('ignore')
```

.. code:: ipython3

```
def min_max(x, axis=None):
    min = x.min(axis=axis, keepdims=True)
    max = x.max(axis=axis, keepdims=True)
    result = (x-min)/(max-min)
    return result

def normalize_c(data):
    for i in range(len(data[0])):
        data[:,i]=min_max(data[:,i])
    return data
```

.. code:: ipython3

```
csv_file = 'Supplementary Data.csv'
with open(csv_file, 'r') as f:
    lines = f.readlines()
```

```
    separator_idx = next(i for i, line in enumerate(lines) if line.strip()
== '')

    mag_part = pd.read_csv(StringIO(''.join(lines[:separator_idx])))
    xy_part = pd.read_csv(StringIO(''.join(lines[separator_idx + 1:])))

    current = mag_part['current'].to_numpy()
    freq_labels = mag_part.columns[1:]
    freq = np.array([float(label.replace('GHz', ''))  for  label  in
freq_labels])
    mag1 = mag_part.drop(columns='current').to_numpy()

    t_current = [xy_part[f'X{i}'].dropna().tolist() for i in range(8)]
    t_freq = [xy_part[f'Y{i}'].dropna().tolist() for i in range(8)]
```

```
.. code:: ipython3

    plt.figure(figsize=(10, 6))
    plt.pcolormesh(current, freq, mag1.T, shading='auto', cmap='viridis')
    plt.colorbar(label='Magnitude')
    plt.xlabel('Current [mA]')
    plt.ylabel('Frequency [GHz]')
    plt.tight_layout()
    plt.show()

    plt.figure(figsize=(8, 6))
    for i in range(8):
        plt.plot(t_current[i], t_freq[i], marker='o', label=f'Band {i}')
    plt.xlabel('Current [mA]')
    plt.ylabel('Frequency [GHz]')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

.. code:: ipython3

```
    def QbHoHami_omegaCsz(N,eps, g1, g2, d1, d2, omegaC0,eps2,A,L,R):
        sz1=tensor(qeye(2),sigmaz(),qeye(N))
        sx1=tensor(qeye(2),sigmax(),qeye(N))
        sz2=tensor(sigmaz(),qeye(2),qeye(N))
        sx2=tensor(sigmax(),qeye(2),qeye(N))
        a=tensor(qeye(2),qeye(2),destroy(N))
        if eps>0:
            omega=omegaC0*(1+R*eps)
        else:
            omega=omegaC0*(1-L*eps)
        Hq2=(eps2+A*eps)*sz2+d2*sx2+g2*sz2*(a.dag()+a)
        Hq1=eps*sz1+d1*sx1-g1*sz1*(a.dag()+a)
        Hr=omega*a.dag()*a
        totalHami =Hq1+Hq2+Hr-2*g1*g2/omega*sz1*sz2
        return totalHami

    def compute(flist,params):
        F=13
        idx = 0
        evals_mat = np.zeros((len(flist),2*2*F))
        for f in (flist):

H=QbHoHami_omegaCsz(F,2*(f+params[8])*params[5],params[0],params[1],params[
2],params[3],params[4],params[6],params[7],params[9],params[10])
            evals, ekets = H.eigenstates()
            evals_mat[idx,:] = np.real(evals)
            idx += 1
        return evals_mat
```

.. code:: ipython3

```
    init_params=(   3.32307555e+00,       3.42675355e+00,       6.47361336e-01,
6.30739015e-01,      5.14723380e+00,      5.04033563e+01,   -1.60345489e+00,   -
9.42530426e-03, 5.46762524e-01,  7.26744297e-04,  7.82749406e-04)
    current1=np.linspace(-0.61,-0.48,10001)
    E1Q=compute(current1,init_params)

.. code:: ipython3

    fig, ax = plt.subplots(figsize=(9,7))
    plt.rcParams["font.size"] = 19
    plt.rcParams["font.family"] ="Times New Roman"
    plt.rcParams["mathtext.fontset"] ="stix"


    for i in range(len(t_current)):
        ax.plot(t_current[i],t_freq[i],"+",label=str(i))
    ax.plot(current1,                 np.array(E1Q[:,5])-np.array(E1Q[:,1]),
'-.',color='b',label='')
    ax.plot(current1,                 np.array(E1Q[:,2])-np.array(E1Q[:,0]),
'-.',color='black',label='' )
    ax.plot(current1,                 np.array(E1Q[:,3])-np.array(E1Q[:,1]),
'-.',color='g',label='' )
    ax.plot(current1,                 np.array(E1Q[:,4])-np.array(E1Q[:,0]),
'-.',color='r',label='' )
    ax.plot(current1,                 np.array(E1Q[:,6])-np.array(E1Q[:,2]),
'-.',color='orange',label='' )
    ax.plot(current1,                 np.array(E1Q[:,3])-np.array(E1Q[:,0]),
'-.',color='yellow',label='' )
    ax.plot(current1,                 np.array(E1Q[:,4])-np.array(E1Q[:,1]),
'-.',color='pink',label='' )
    ax.plot(current1,                 np.array(E1Q[:,5])-np.array(E1Q[:,0]),
'-.',color='gray',label='' )

    ax.set_ylim([4.7,6])
    # ax.set_ylim([5.2,5.4])
```

```
    plt.grid()
    plt.xlabel(r"mA",fontsize=24)
    plt.ylabel(r"$E/h¥ $[GHz]",fontsize=22)
    plt.legend(loc="upper right",ncol=1)
    # plt.savefig("Large dispersive readout.png",dpi=700)
    plt.show()
```

.. code:: ipython3

```
    def err(params):

        Ef=[[] for i in range(len(t_current))]
        for i in tqdm([0,1,2,3,4,6,7]):
            Ef[i]=compute(t_current[i],params)

        ere1=abs(t_freq[0] - np.array(Ef[0][:,2])+np.array(Ef[0][:,0]))
        ere2=abs(t_freq[1] - np.array(Ef[1][:,3])+np.array(Ef[1][:,0]))
        ere3=abs(t_freq[2] - np.array(Ef[2][:,4])+np.array(Ef[2][:,0]))
        ere4=abs(t_freq[3] - np.array(Ef[3][:,4])+np.array(Ef[3][:,1]))
        ere5=abs(t_freq[4] - np.array(Ef[4][:,5])+np.array(Ef[4][:,0]))
        # ere6=abs(t_freq[5] - np.array(Ef[5][:,5])+np.array(Ef[5][:,1]))
        ere7=abs(t_freq[6] - np.array(Ef[6][:,4])+np.array(Ef[6][:,0]))
        ere8=abs(t_freq[7] - np.array(Ef[7][:,3])+np.array(Ef[7][:,0]))

err=100*[np.sum(ere1)/80,20*np.sum(ere2)/35,10*np.sum(ere3)/11,np.sum(ere4)
/2,np.sum(ere5)/5,0,sum(ere7)/14,sum(ere8)/14,0,0,0,0]

        return err
    Ere=err(init_params)
    print(np.sum(Ere))
```

.. code:: ipython3

```
    start = time.time()

    fitparams=leastsq(err,init_params)

    elapsed_time = time.time() - start
    print ("elapsed_time:{0}".format(elapsed_time/60) + "[min]")
```

.. code:: ipython3

```
    #   fitparams=[   3.33492946e+00,      3.44534894e+00,      6.54174620e-01,
6.34830707e-01,
    #    5.14518205e+00,   5.06046239e+01, -1.60772755e+00,   2.19593250e-03,
    #    5.46796238e-01,   7.50660250e-04,   8.95914996e-04]
    fitparams=fitparams[0]
    print(fitparams)
    E1Q=compute(current1,fitparams)
    print(np.sum(err(fitparams)))
```

.. code:: ipython3

```
    D1=fitparams[2]*np.exp(-2*(fitparams[0]/fitparams[4])**2)
    D2=fitparams[3]*np.exp(-2*(fitparams[1]/fitparams[4])**2)
    fitparams_0=[0,   0,   D1,   D2,     fitparams[4]*1.0317,     fitparams[5],
fitparams[6]*0.973,       fitparams[7],       fitparams[8],       fitparams[9],
fitparams[10]]
    E1Q0=compute(current1,fitparams_0)
```

.. code:: ipython3

```
    eps0=4*(current+fitparams[8])*fitparams[5]
    eps=4*(current1+fitparams[8])*fitparams[5]
```

.. code:: ipython3

```
    plt.rcParams["font.size"] = 20
    plt.rcParams["font.family"] ="Arial"
    plt.rcParams["mathtext.fontset"] ="stix"
```

.. code:: ipython3

```
    fig, ax = plt.subplots(figsize=(10,12))

    X, Y = np.meshgrid(freq,eps0)
    mappable =ax.pcolor(Y,X,min_max(normalize_c(mag1)), cmap="summer")
    cbar_num_format = "%.2f"
    cbar = plt.colorbar(mappable,ax=ax, format=cbar_num_format)
    cbar.ax.set_yticklabels(['0','0.2','0.4','0.6','0.8','1'])

    ax.plot(eps,                          np.array(E1Q[:,3])-np.array(E1Q[:,0]),
'-.',lw=3,color='b',label='$¥omega_{30}$' )
    ax.plot(eps,                          np.array(E1Q[:,4])-np.array(E1Q[:,0]),
'-.',lw=3,color='r',label='$¥omega_{40}$' )
    ax.plot(eps,                          np.array(E1Q[:,5])-np.array(E1Q[:,1]),
'-.',lw=3,color='black',label='$¥omega_{51}$')



    plt.vlines([-1.2*2],5.201,5.399,  "purple", linestyles='--')
    plt.vlines([-0.8*2], 5.201,5.399,  "purple", linestyles='--')
    plt.hlines([5.201], -1.2*2,-0.8*2,  "purple", linestyles='--')
    plt.hlines([5.399], -1.2*2,-0.8*2,  "purple", linestyles='--')

    ax.set_xlim([-5,5])
    ax.set_ylim([5.2,5.4])
    plt.yticks([5.2, 5.3, 5.4])
```

```
    plt.xticks([-5,-2.5,0,2.5,5])
    plt.xlabel(r"$¥varepsilon_1$ [GHz]",fontsize=20)
    plt.ylabel(r"$¥omega_p¥ $[GHz]",fontsize=20)
    plt.legend(loc="lower                                                left",ncol=1,
labelcolor='black',facecolor="whitesmoke",framealpha=0.5)
    #                   plt.savefig("1R2QAS_enlarge.png",bbox_inches="tight",
pad_inches=0.5,dpi=700)
    plt.show()
```

```
.. code:: ipython3

    fig, ax = plt.subplots(figsize=(12.5,12))

    X, Y = np.meshgrid(freq,eps0)
    mappable =ax.pcolor(Y,X,min_max(normalize_c(mag1)), cmap="summer")
    cbar_num_format = "%.2f"
    cbar = plt.colorbar(mappable,ax=ax, format=cbar_num_format)
    cbar.ax.set_yticklabels(['0','0.2','0.4','0.6','0.8','1'])

    ax.plot(eps,                        np.array(E1Q[:,2])-np.array(E1Q[:,0]),
'-.',lw=3,color='magenta',label=r'$¥omega_{20}$' )
    ax.plot(eps,                        np.array(E1Q[:,3])-np.array(E1Q[:,0]),
'-.',lw=3,color='blue',label=r'$¥omega_{30}$' )
    ax.plot(eps,                        np.array(E1Q[:,3])-np.array(E1Q[:,1]),
'-.',lw=3,color='purple',label=r'$¥omega_{31}$' )
    ax.plot(eps,                        np.array(E1Q[:,4])-np.array(E1Q[:,0]),
'-.',lw=3,color='r',label=r'$¥omega_{40}$' )
    ax.plot(eps,                        np.array(E1Q[:,4])-np.array(E1Q[:,1]),
'-.',lw=3,color='yellow',label=r'$¥omega_{41}$' )
    ax.plot(eps,                        np.array(E1Q[:,5])-np.array(E1Q[:,0]),
'-.',lw=3,color='cyan',label=r'$¥omega_{50}$' )
    ax.plot(eps,                        np.array(E1Q[:,5])-np.array(E1Q[:,1]),
```

```
'-.',lw=3,color='black',label=r'$¥omega_{51}$')


    plt.vlines([-5],5.2,5.4,  "red", linestyles='--')
    plt.vlines([5], 5.2,5.4,  "red", linestyles='--')
    plt.hlines([5.2], -5,5,  "red", linestyles='--')
    plt.hlines([5.4], -5,5,  "red", linestyles='--')


    ax.set_xlim([-12,12])
    ax.set_ylim([4.7,6])
    plt.yticks([4.8,5,5.2,5.4,5.6,5.8,6])
    # plt.grid()
    plt.xlabel(r"$¥varepsilon_1$ [GHz]",fontsize=20)
    plt.ylabel(r"$¥omega_p¥ $[GHz]",fontsize=20)
    plt.legend(bbox_to_anchor=(1,  1),  loc='upper  right',  borderaxespad=0,
labelspacing=0.2,                                                    ncol=1,
labelcolor='black',facecolor="whitesmoke",framealpha=0.5)
    #                    plt.savefig("1R2QAS_withg0.png",bbox_inches="tight",
pad_inches=0.5,dpi=700)
    plt.show()
```

```
.. code:: ipython3

    fig, ax = plt.subplots(figsize=(10,7))

    X, Y = np.meshgrid(freq,eps0)
    mappable =ax.pcolor(Y,X,min_max(normalize_c(mag1)), cmap="summer")
    cbar_num_format = "%.2f"

    ax.plot(eps,          np.array(E1Q0[:,3])-np.array(E1Q0[:,0]),          '-
',lw=2,color='white' )
```

```
    ax.plot(eps,          np.array(E1Q0[:,4])-np.array(E1Q0[:,0]),          '-
',lw=2,color='white' )
    ax.plot(eps,                        np.array(E1Q[:,3])-np.array(E1Q[:,0]),
'-.',lw=3,color='b',label='$¥omega_{30}$' )
    ax.plot(eps,                        np.array(E1Q[:,4])-np.array(E1Q[:,0]),
'-.',lw=3,color='r',label='$¥omega_{40}$' )

    ax.set_xlim([-2.4,-1.6])
    ax.set_ylim([5.2,5.4])
    plt.yticks([5.2, 5.3, 5.4])
    ax.axes.xaxis.set_visible(False)
    plt.ylabel(r"$¥omega_p $[GHz]",fontsize=25)
    plt.legend(loc="lower                                              left",ncol=1,
labelcolor='black',facecolor="whitesmoke",framealpha=0.5)
    #                 plt.savefig("1R2QAS_enlarge2.png",bbox_inches="tight",
pad_inches=0.5,dpi=700)
    plt.show()
```

```
.. code:: ipython3

    fig, ax = plt.subplots(figsize=(8,10))

    X, Y = np.meshgrid(freq,eps0)
    mappable =ax.pcolor(Y,X,min_max(normalize_c(mag1)), cmap="summer")
    cbar_num_format = "%.2f"

    ax.set_xlim([-12,12])
    ax.set_ylim([4.7,6])
    plt.yticks([4.8,5,5.2,5.4,5.6,5.8,6])
    plt.xlabel(r"$¥varepsilon_1$ [GHz]",fontsize=20)
    plt.ylabel(r"$¥omega_p¥ $[GHz]",fontsize=20)
    #                       plt.savefig("1R2QAS_raw.png",bbox_inches="tight",
```

```
pad_inches=0.5,dpi=700)
    plt.show()
```

```
.. code:: ipython3

    xp=eps[np.argmin(np.array(E1Q[:,4])-np.array(E1Q[:,3]))]

    fig, ax = plt.subplots(figsize=(6,8))

    ax.plot(eps,            np.array(E1Q[:,1])-np.array(E1Q[:,0]),          '-
',label='$¥omega_{01}$' )
    ax.plot(eps,            np.array(E1Q[:,2])-np.array(E1Q[:,0]),          '-
',color='green',label='$¥omega_{20}$' )
    ax.plot(eps,            np.array(E1Q[:,3])-np.array(E1Q[:,0]),          '-
',color='b',label='$¥omega_{30}$' )
    ax.plot(eps,            np.array(E1Q[:,4])-np.array(E1Q[:,0]),          '-
',color='r',label='$¥omega_{40}$' )

    plt.vlines([-2.2*0.9999],5.1,5.5,  "purple", linestyles='--')
    plt.vlines([-1.8], 5.1,5.5,  "purple", linestyles='--')
    plt.hlines([5.1], -2.2,-1.8,  "purple", linestyles='--')
    plt.hlines([5.5], -2.2,-1.8,  "purple", linestyles='--')

    plt.axvline(x=xp,ymin=0, ymax=10,color='black',linestyle='--')

    ax.set_xlim([-2.2,-1.8])
    ax.set_ylim([1,6])
    plt.grid()

    plt.text(-2.05, 3, r"$¥omega_{02}=$3.21 GHz" )
    plt.text(-2.02, 1.65, r"$¥omega_{01}=$2.11 GHz" )
```

```
    plt.xlabel(r"$¥varepsilon_1$ [GHz]",fontsize=20)
    plt.ylabel(r"$¥omega_{ij}¥ $[GHz]",fontsize=20)
    plt.legend(bbox_to_anchor=(0.5,0.43),                                    ncol=1,
labelcolor='black',facecolor="whitesmoke",framealpha=0.5)
    # plt.savefig("wq12wr.png",bbox_inches="tight", pad_inches=0.5,dpi=700)
    plt.show()
```

```
.. code:: ipython3

    fig, ax = plt.subplots(figsize=(6,5))

    ax.plot(eps,          np.array(E1QO[:,3])-np.array(E1QO[:,0]),          '-
',lw=2,color='black' )
    ax.plot(eps,          np.array(E1QO[:,4])-np.array(E1QO[:,0]),          '-
',lw=2,color='black' )
    ax.plot(eps,           np.array(E1Q[:,3])-np.array(E1Q[:,0]),           '-
',color='b',label='$¥omega_{30}$' )
    ax.plot(eps,           np.array(E1Q[:,4])-np.array(E1Q[:,0]),           '-
',color='r',label='$¥omega_{40}$' )

    ax.set_xlim([-2.2,-1.8])
    ax.set_ylim([5.1,5.5])
    plt.text(-1.98, 5.26, r"5.31 GHz" )
    plt.xlabel(r"$¥varepsilon_1$ [GHz]",fontsize=20)
    plt.ylabel(r"$¥omega_{ij}¥ $[GHz]",fontsize=20)
    plt.legend(loc="upper                    right",                    ncol=1,
labelcolor='black',facecolor="whitesmoke",framealpha=0.5)
    #                  plt.savefig("wq12wr_enlarge.png",bbox_inches="tight",
pad_inches=0.1,dpi=700)
    plt.show()
```

.. code:: ipython3

```
    def QbHoHami_omegaCsz(N,eps, g1, g2, d1, d2, omegaC0,eps2,A,L,R):
        N=13
        sz2=tensor(qeye(2),sigmaz(),qeye(N))
        sx2=tensor(qeye(2),sigmax(),qeye(N))
        sz1=tensor(sigmaz(),qeye(2),qeye(N))
        sx1=tensor(sigmax(),qeye(2),qeye(N))
        a=tensor(qeye(2),qeye(2),destroy(N))
        if eps>0:
            omega=omegaC0*np.sqrt(1+R*eps)
        if eps<=0:
            omega=omegaC0*np.sqrt(1-L*eps)
        if eps==0:
            th1=np.pi/2
            eps1=np.sqrt(eps**2+d1**2)
        if eps!=0:
            th1=np.arctan(-d1/abs(eps))
            eps1=np.sqrt(eps**2+d1**2)
        th2=np.arctan(-d2/abs(eps2+A*eps))
        L1=g1*(sz1*np.cos(th1)+sx1*np.sin(th1))
        L2=g2*(sz2*np.cos(th2)+sx2*np.sin(th2))
        Hq2=np.sign(eps2)*np.sqrt((eps2+A*eps)**2+d2**2)*sz2
        Hq1=np.sign(eps)*eps1*sz1
        Hr=omega*a.dag()*a
        totalHami =Hq1+Hq2+Hr-2*L1*L2/omega+(L1-L2)*(a.dag()+a)
        return totalHami

    #Rabimodel Hamiltonian for fit
    def compute(flist,params):
        F=13
        idx = 0
        evals_mat = np.zeros((len(flist),2*2*F))
```

```
        eket=[]
        for f in (flist):

H=QbHoHami_omegaCsz(F,2*(f+params[8])*params[5],params[0],params[1],params[
2],params[3],params[4],params[6],params[7],params[9],params[10])
            evals, ekets = H.eigenstates()
            evals_mat[idx,:] = np.real(evals)
            eket.append(ekets)
            idx += 1
        return evals_mat,eket
```

```
.. code:: ipython3

    init_params=[   3.33492946e+00,       3.44534894e+00,       6.54174620e-01,
6.34830707e-01,
                    5.14518205e+00,      5.06046239e+01,      -1.60772755e+00     ,
2.19593250e-03,
                    5.46796238e-01 ,  7.50660250e-04 ,  8.95914996e-04]
    wr=init_params[4]
    g2=init_params[1]

    current1=np.linspace(-0.61,-0.48,10001)
    e2,e2ket=compute(current1,init_params)
```

```
.. code:: ipython3

    D1=init_params[2]*np.exp(-2*(init_params[0]/init_params[4])**2)
    D2=init_params[3]*np.exp(-2*(init_params[1]/init_params[4])**2)
```

```
.. code:: ipython3

    init_params_g0=[0,  0,  D1,  D2,   init_params[4]*1.0317,   init_params[5],
init_params[6]*0.973,    init_params[7],    init_params[8],    init_params[9],
init_params[10]]
    eg0,eg0ket=compute(current1,init_params_g0)
```

.. code:: ipython3

```
elist=4*(current1+init_params[8])*init_params[5]
```

.. code:: ipython3

```
plt.rcParams["font.size"] = 20
plt.rcParams["font.family"] ="Arial"
plt.rcParams["mathtext.fontset"] ="stix"
```

.. code:: ipython3

```
fig, ax = plt.subplots(figsize=(8,8))


ax.plot(elist/wr, (eg0[:,1]-eg0[:,0])/wr,'-',lw=2,color='pink')
ax.plot(elist/wr, (eg0[:,2]-eg0[:,0])/wr,'-',lw=2,color='pink')
ax.plot(elist/wr, (eg0[:,3]-eg0[:,0])/wr,'-',lw=2,color='pink')
ax.plot(elist/wr, (eg0[:,4]-eg0[:,0])/wr,'-',lw=2,color='pink')
ax.plot(elist/wr, (eg0[:,5]-eg0[:,0])/wr,'-',lw=2,color='pink')
ax.plot(elist/wr, (eg0[:,6]-eg0[:,0])/wr,'-',lw=2,color='pink')

ax.plot(elist/wr,                                    (e2[:,1]-e2[:,0])/wr,'-
',color='green',label=r"$¥omega_{10}$")
ax.plot(elist/wr,                                    (e2[:,2]-e2[:,0])/wr,'-
',color='black',label=r"$¥omega_{20}$")
ax.plot(elist/wr,                                    (e2[:,3]-e2[:,0])/wr,'-
',color='blue',label=r"$¥omega_{30}$")
ax.plot(elist/wr,                                    (e2[:,4]-e2[:,0])/wr,'-
',color='red',label=r"$¥omega_{40}$")
ax.plot(elist/wr,                                    (e2[:,5]-e2[:,0])/wr,'-
',color='cyan',label=r"$¥omega_{50}$")
ax.plot(elist/wr,                                    (e2[:,6]-e2[:,0])/wr,'-
',color='orange',label=r"$¥omega_{60}$")

ax.legend(bbox_to_anchor              =              (-0.02,1.02),loc="upper
```

```
left",ncol=3,fontsize=20)
    ax.set_xlim([-2,2])
    ax.set_ylim([0.0,2.5])
    plt.yticks([0,1,2],['0','1','2'])
    ax.set_xlabel(r'$¥varepsilon_{1}/¥omega_¥mathrm{r}$')
    ax.set_ylabel(r'$¥omega_{i0}/¥omega_¥mathrm{r}$')
    plt.grid()
    plt.savefig('g0overwrite_d_wr_e2',                    bbox_inches="tight",
pad_inches=0.5,dpi=700)
```

```
.. code:: ipython3

    A=3
    B=4
```

```
.. code:: ipython3

    fig, ax = plt.subplots(figsize=(8,8))

    ax.plot(elist, (eg0[:,1]-eg0[:,0]),'-',lw=1,color='pink')
    ax.plot(elist, (eg0[:,2]-eg0[:,0]),'-',lw=1,color='pink')
    ax.plot(elist, (eg0[:,3]-eg0[:,0]),'-',lw=1,color='pink')
    ax.plot(elist, (eg0[:,4]-eg0[:,0]),'-',lw=1,color='pink')
    ax.plot(elist, (eg0[:,5]-eg0[:,0]),'-',lw=1,color='pink')
    ax.plot(elist, (eg0[:,6]-eg0[:,0]),'-',lw=1,color='pink')

    ax.plot(elist,                                    (e2[:,3]-e2[:,0]),'-
',color='blue',label=r"$¥omega_{30}$")
    ax.plot(elist,                                    (e2[:,4]-e2[:,0]),'-
',color='red',label=r"$¥omega_{40}$")

    ax.set_xlim([-2.2,-1.8])
```

```
    ax.set_ylim([5.1,5.5])
    ax.set_xlabel(r'$¥varepsilon_{1}$ [GHz]')
    ax.set_ylabel(r'$¥omega_{ij}$ [GHz]')
    plt.grid()
    plt.savefig('g0overwrite_d_wr_e2',                    bbox_inches="tight",
pad_inches=0.5,dpi=700)
```

```
.. code:: ipython3

    gg1=[]
    ee0=[]
    gg1a=[]
    ee0a=[]
    N=13
    a=tensor(qeye(2),qeye(2),destroy(N))

    for t in range(len(elist)):
        s_gg1=tensor(basis(2,0),basis(2,0),basis(N,1))
        s_ee0=tensor(basis(2,1),basis(2,1),basis(N,0))

        ketB=e2ket[t][B]
        ketA=e2ket[t][A]
        gg1_exp=(ketB).dag()*s_gg1
        ee0_exp=(ketB).dag()*s_ee0

        gg1a_exp=(ketA).dag()*s_gg1
        ee0a_exp=(ketA).dag()*s_ee0

        gg1.append((gg1_exp**2).real)
        ee0.append((ee0_exp**2).real)
        gg1a.append((gg1a_exp**2).real)
        ee0a.append((ee0a_exp**2).real)
```

.. code:: ipython3

```
    fig, ax = plt.subplots(ncols=1, figsize=(10,4))

    ax.plot(elist,gg1,lw=3,color='b',label="$P_{gg1}^{(3)}$")
    ax.plot(elist,ee0,'-.',lw=3,color='b',label="$P_{ee0}^{(3)}$")

    ax.plot(elist,gg1a,'-',lw=3,color='r',label="$P_{gg1}^{(4)}$")
    ax.plot(elist,ee0a,'-.',lw=3,color='r',label="$P_{ee0}^{(4)}$")

    ax.grid()
    ax.set_ylim([0,0.9])
    ax.set_xlim([-2.4,-1.6])
    plt.xticks([-2.4,-2.2,-2,-1.8,-1.6],['-2.4','-2.2','-2','-1.8','-1.6'])
    plt.yticks([0,0.2,0.4,0.6,0.8],['0','0.2','0.4','0.6','0.8'])
    ax.legend(loc="lower left",ncol=1, fontsize=20)
    ax.set_xlabel(r'$¥varepsilon_{1}$ [GHz]', fontsize=20)
    ax.set_ylabel('Expectation value',fontsize=20)
    plt.savefig('State03_04AS', bbox_inches="tight", pad_inches=0.5,dpi=700,
transparent=True)
```

.. code:: ipython3

```
    elist0=np.linspace(-1*wr,-0.0*wr,201)
    c_list=elist0/4/init_params[5]-init_params[8]

    gw_list=np.linspace(0,1,201)

    s_gg1=tensor(basis(2,0),basis(2,0),basis(N,1))
    s_ee0=tensor(basis(2,1),basis(2,1),basis(N,0))

    Pgg1=[]
    Pee0=[]
```

```
    gap=[]
    for gw in tqdm(gw_list):
        init_params[0]=gw*wr
        init_params[1]=gw*wr
        evals_mat1,ekets_mat1 = compute(c_list,init_params)
        eA0=evals_mat1[:,A]-evals_mat1[:,0]
        eB0=evals_mat1[:,B]-evals_mat1[:,0]
        e0=c_list[np.argmin(abs(eA0-eB0))]

        wlist2=np.linspace(e0-
0.01*init_params[8],e0+0.01*init_params[8],5001)
        evals_mat2,ekets_mat2 = compute(wlist2,init_params)

        eA=evals_mat2[:,A]-evals_mat2[:,0]
        eB=evals_mat2[:,B]-evals_mat2[:,0]
        gap.append(np.min(abs(eA-eB))/2)
        t=np.argmin(abs(eA-eB))

        eket_q1=[]
        eket_q2=[]

        ketB=ekets_mat2[t][B]
        ketA=ekets_mat2[t][A]

        a3=(ketA.dag()*s_gg1/np.sqrt(2))
        b3=(ketA.dag()*s_ee0/np.sqrt(2))
        a4=(ketB.dag()*s_gg1/np.sqrt(2))
        b4=(ketB.dag()*s_ee0/np.sqrt(2))

        if np.sign(a3)==np.sign(b3):
            a3=abs(a3)
            b3=abs(b3)
        if np.sign(a3)!=np.sign(b3):
            a3=abs(a3)
            b3=-abs(b3)
```

```
        if np.sign(a4)==np.sign(b4):
            a4=abs(a4)
            b4=abs(b4)
        if np.sign(a4)!=np.sign(b4):
            a4=abs(a4)
            b4=-abs(b4)

        gg1_exp=a3+a4
        ee0_exp=b3-b4

        Pgg1.append((gg1_exp**2).real)
        Pee0.append((ee0_exp**2).real)
```

```
.. code:: ipython3

    fig, ax = plt.subplots(figsize=(8,5))

    ax.plot(gw_list[30:200],Pgg1[30:200],'-
',lw=2,color='b',label="$P_{gg1}$")
    ax.plot(gw_list[30:200],Pee0[30:200],'-
',lw=2,color='r',label="$P_{ee0}$")

    ax.legend(loc="upper right",ncol=2,fontsize=20)
    ax.set_xlim([0.15,1])
    ax.set_ylim([0.75,1])
    plt.xticks([0.25,0.5,0.75,1],['0.25','0.5','0.75','1'])
    plt.yticks([0.8,0.9,1],['0.8','0.9','1'])
    plt.vlines(0.67, 0.6, 1, color='g', linestyles='dotted')
    ax.set_xlabel(r'$g/¥omega_¥mathrm{r}$')
    ax.set_ylabel('Ocupation probability')
    plt.grid()
    plt.savefig('maxP', bbox_inches="tight", pad_inches=0.2,dpi=700)
```

.. code:: ipython3

```
gs=np.argmin(abs(0.64-gw_list))
gs2=np.argmin(abs(g2/wr-gw_list))

fig, ax = plt.subplots(figsize=(8,5))

ax.plot(gw_list, np.array(gap)*1e3,'-',color='black')
ax.plot([gw_list[gs]],[gap[gs]*1e3],'*',color='red',markersize='15')
ax.plot([gw_list[gs2]],[gap[gs2]*1e3],'*',color='blue',markersize='15')

ax.set_xlim([0,1])
# ax.set_ylim([0,30])
plt.xticks([0,0.25,0.5,0.75,1],['0','0.25','0.5','0.75','1'])

ax.set_xlabel(r'$g/¥omega_¥mathrm{r}$')
ax.set_ylabel(r'$g_¥mathrm{30-40}$[MHz]')
plt.grid()
plt.savefig('geff', bbox_inches="tight", pad_inches=0.5,dpi=700)
```