■ Concepts and Methods of 2D Infrared Spectroscopy

Peter Hamm and Martin T.Zanni

This progam was used to produce Fig. 7.8

Define response functions for rephasing and non-rephasing diagrams for a three-level system (Equ. 7.40)

```
In[1]:= Rnr = (Exp[-I (t3 + t1) w] - Exp[-I (t3 (w - Δ) + t1 w)]) *
            Exp[-g[t1] - g[t2] - g[t3] + g[t1 + t2] + g[t2 + t3] - g[t1 + t2 + t3]];
        Rr = (Exp[-I (t3 - t1) w] - Exp[-I (t3 (w - Δ) - t1 w)]) *
            Exp[-g[t1] + g[t2] - g[t3] - g[t1 + t2] - g[t2 + t3] + g[t1 + t2 + t3]];
```

Define Kubo-lineshape-function (Eq. 7.25)

```
In[3]:= g[t_] = Δw^2 τc^2 (Exp[-t / τc] + t / τc - 1);
```

Parameters are in units of ps and ps-1; we set the center frequency to w=0, which is equivalent to a measurement in the rotating frame

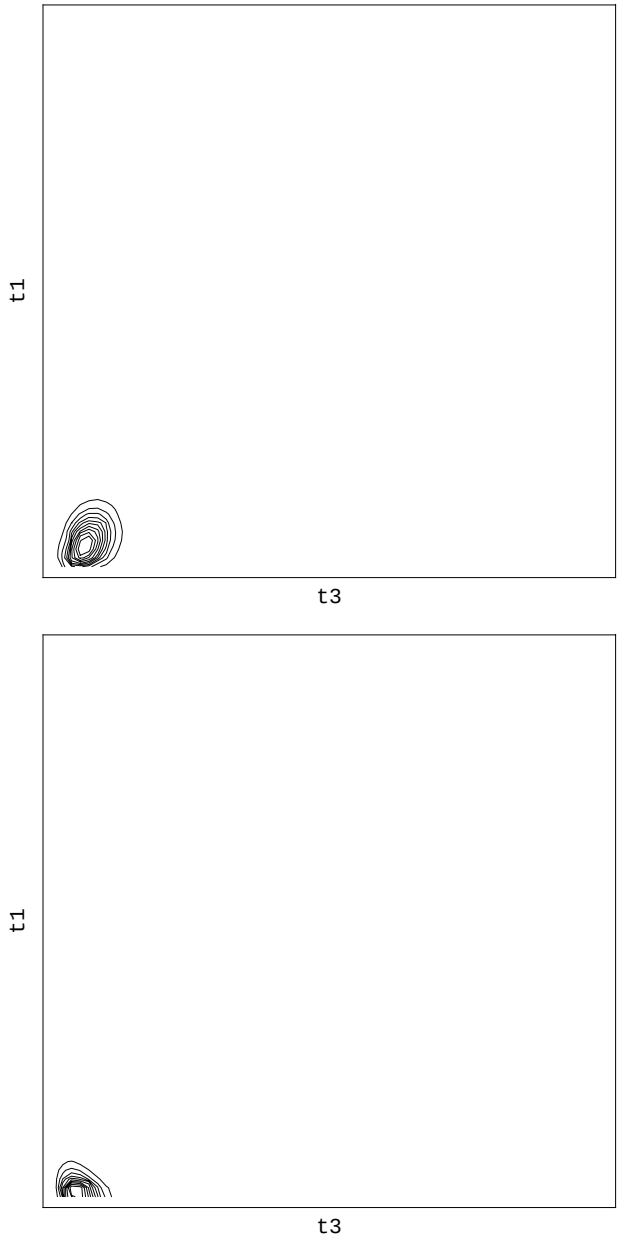
```
In[4]:= Δ = 5;
        Δw = 5;
        τc = 1;
        w = 0;
```

Collect rephasing and non-rephasing data on a grid with stepsize dt and nt data points with population time t2. The first time-point needs to be halved (Sect. 9.5.3).

```
In[8]:= t2 = .1;
        nt = 64;
        dt = .1;
        Rnrlist = Table[Rnr, {t1, 0, (nt - 1) * dt, dt}, {t3, 0, (nt - 1) * dt, dt}];
        Rrlist = Table[Rr, {t1, 0, (nt - 1) * dt, dt}, {t3, 0, (nt - 1) * dt, dt}];
        For[i = 1, i ≤ nt, i++, Rnrlist[[i, 1]] /= 2; Rrlist[[i, 1]] /= 2];
        For[i = 2, i ≤ nt, i++, Rnrlist[[1, i]] /= 2; Rrlist[[1, i]] /= 2];
```

Plot time-domain data

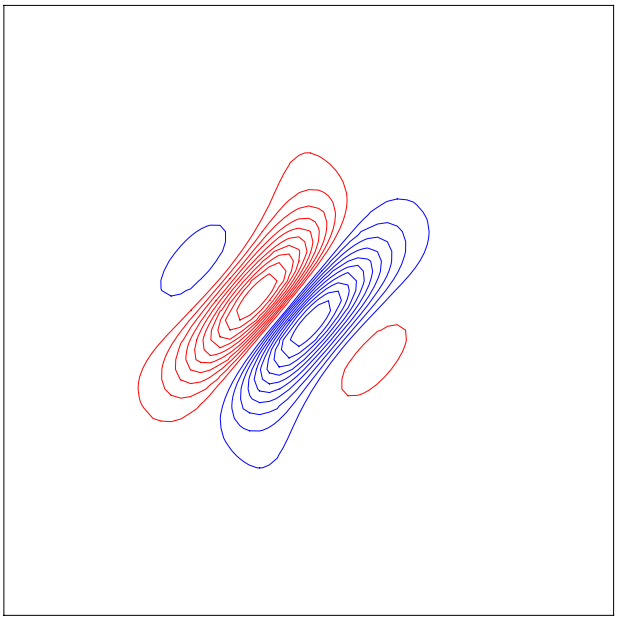
```
In[15]:= ListContourPlot[Re[Rrlist], PlotRange → All, Contours → 10,
           ContourShading → False, FrameTicks → None, FrameLabel → {t3, t1}];
         ListContourPlot[Re[Rnrlist], PlotRange → All, Contours → 10,
           ContourShading → False, FrameTicks → None, FrameLabel → {t3, t1}];
```



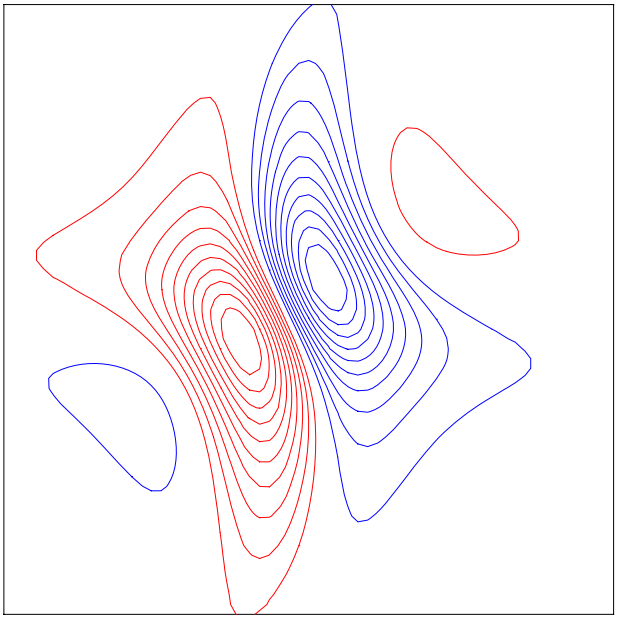
Perform 2D Fourier transform and re-order data so that w1=w3=0 is centerd in the middle. Frequency axis w1 is inverted.

```
In[17]:= spectrum2Dr = Fourier[Rrlist];
         spectrum2Dr = Reverse[Drop[RotateRight[spectrum2Dr, {nt / 2, nt / 2}], 1, 1]];

         spectrum2Dnr = Fourier[Rnrlist];
         spectrum2Dnr = Drop[RotateRight[spectrum2Dnr, {nt / 2, nt / 2}], 1, 1];

         spectrum2Dabs = Re[spectrum2Dr + spectrum2Dnr];
```

In[22]:=
```
max = Max[Max[Re[spectrum2Dr]], -Min[Re[spectrum2Dr]]];
p1 = ListContourPlot[Re[spectrum2Dr],
  PlotRange → {0, max}, ContourShading → False, Contours → 10, Ticks → None,
  ContourStyle → {RGBColor[0, 0, 1]}, DisplayFunction -> Identity];
p2 = ListContourPlot[Re[spectrum2Dr], PlotRange → {-max, 0},
  ContourShading → False, Contours → 10, Ticks → None,
  ContourStyle → {RGBColor[1, 0, 0]}, DisplayFunction -> Identity];
Show[{p1, p2}, PlotRange → {{1, nt - 1}, {1, nt - 1}},
  FrameTicks → {None, None}, DisplayFunction → $DisplayFunction];
```



In[26]:=
```
max = Max[Max[Re[spectrum2Dnr]], -Min[Re[spectrum2Dnr]]];
p1 = ListContourPlot[Re[spectrum2Dnr],
  PlotRange → {0, max}, ContourShading → False, Contours → 10, Ticks → None,
  ContourStyle → {RGBColor[0, 0, 1]}, DisplayFunction -> Identity];
p2 = ListContourPlot[Re[spectrum2Dnr], PlotRange → {-max, 0},
  ContourShading → False, Contours → 10, Ticks → None,
  ContourStyle → {RGBColor[1, 0, 0]}, DisplayFunction -> Identity];
Show[{p1, p2}, PlotRange → {{1, nt - 1}, {1, nt - 1}},
  FrameTicks → {None, None}, DisplayFunction → $DisplayFunction];
```



In[30]:=
```
max = Max[Max[Re[spectrum2Dabs]], -Min[Re[spectrum2Dabs]]];
p1 = ListContourPlot[Re[spectrum2Dabs],
  PlotRange → {0, max}, ContourShading → False, Contours → 10, Ticks → None,
  ContourStyle → {RGBColor[0, 0, 1]}, DisplayFunction -> Identity];
p2 = ListContourPlot[Re[spectrum2Dabs], PlotRange → {-max, 0},
  ContourShading → False, Contours → 10, Ticks → None,
  ContourStyle → {RGBColor[1, 0, 0]}, DisplayFunction -> Identity];
Show[{p1, p2}, PlotRange → {{1, nt - 1}, {1, nt - 1}},
  FrameTicks → {None, None}, DisplayFunction → $DisplayFunction];
```