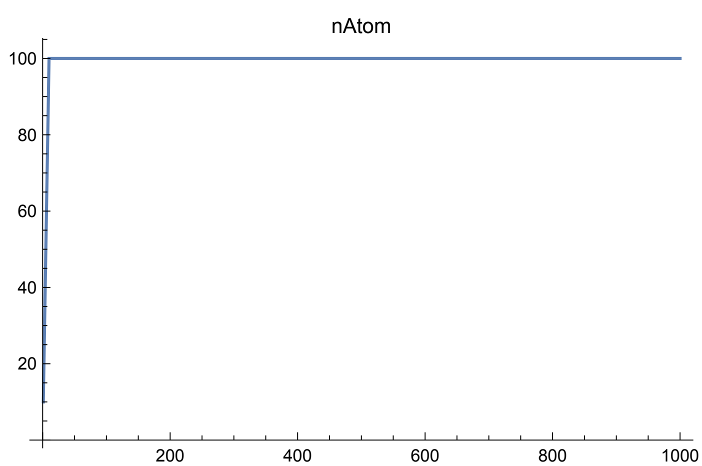
In[419]:=
```
Clear["Global`*"]
```

In[420]:=
```
inputData = Import[NotebookDirectory[] <> "/input.txt"];
input = StringSplit[inputData];
l = Length[input];(*l will have to be even*)
(*Write a function here that will take in strings as variable names*)
dt = Read[StringToStream[input[[2]]]];
tmax = input[[4]];
nTrajectory = input[[6]];
nstore = input[[8]];
yWall = input[[10]];
sigmaXX = input[[12]];
sigmaXZ = input[[14]];
transitTime = input[[16]];
sigmaPX = input[[18]];
sigmaPY = input[[20]];
sigmaPZ = input[[22]];
density = input[[24]];
rabi = input[[26]];
kappa = input[[28]];
lambda = input[[30]];
invT2 = input[[32]];
name = input[[34]]
```

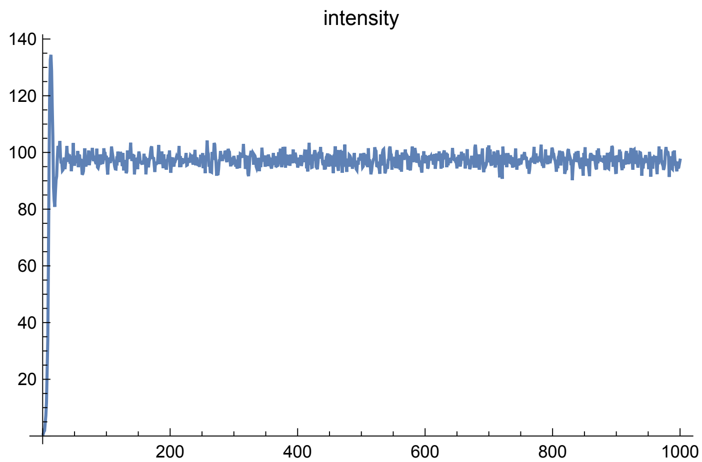
Out[439]=
```
latt_tau1.0_g2_k40_N100_pz0_dz0_REF
```

In[440]:=
```
(*When doing one run*)
intensity = Flatten[Import[NotebookDirectory[] <> name <> "/intensity.dat"]];
nAtom = Flatten[Import[NotebookDirectory[] <> name <> "/nAtom.dat"]];
inversionAve =
  Flatten[Import[NotebookDirectory[] <> name <> "/inversionAve.dat"]];
spinSpinCorAveRe = Flatten[
   Import[NotebookDirectory[] <> name <> "/spinSpinCorAve_re.dat"]];
spinSpinCorAveIm = Flatten[Import[
    NotebookDirectory[] <> name <> "/spinSpinCorAve_im.dat"]];
szFinal = Flatten[Import[NotebookDirectory[] <> name <> "/szFinal.dat"]];
realG1 = Import[NotebookDirectory[] <> name <> "/realG1.dat"];
spectra = Import[NotebookDirectory[] <> name <> "/spectra.dat"];
ListLinePlot[nAtom, PlotRange → All, PlotLabel → "nAtom"]
ListLinePlot[intensity, PlotRange → All, PlotLabel → "intensity"]
ListLinePlot[inversionAve, PlotRange → All, PlotLabel → "inversionAve"]
ListLinePlot[spinSpinCorAveRe, PlotRange → All, PlotLabel → "spinSpinCorAve_re"]
ListLinePlot[spinSpinCorAveIm, PlotRange → All, PlotLabel → "spinSpinCorAve_im"]
ListPlot[szFinal, PlotRange → All, PlotLabel → "inversionOut"]
ListLinePlot[realG1, PlotRange → All, PlotLabel → "realG1"]
ListLinePlot[spectra, PlotRange → All, PlotLabel → "spectra"]
```

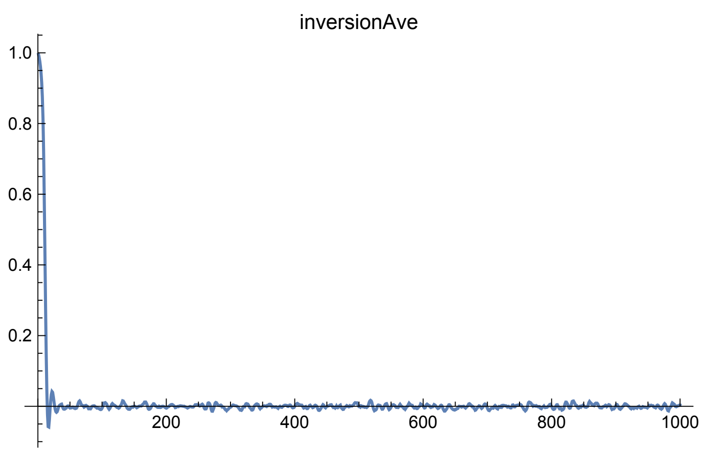
Out[448]=



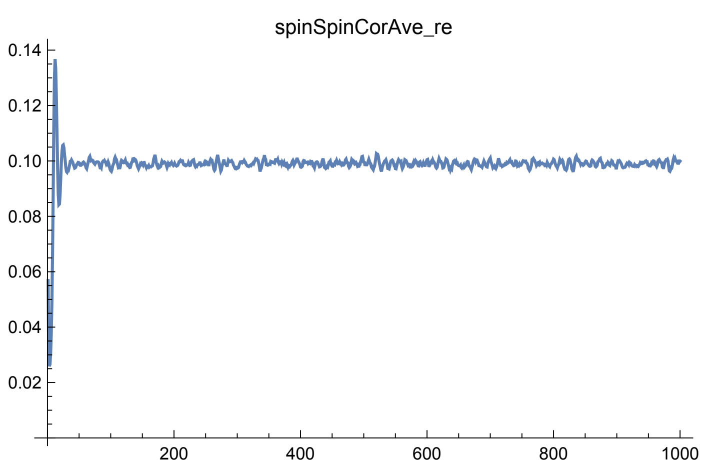
Out[449]=



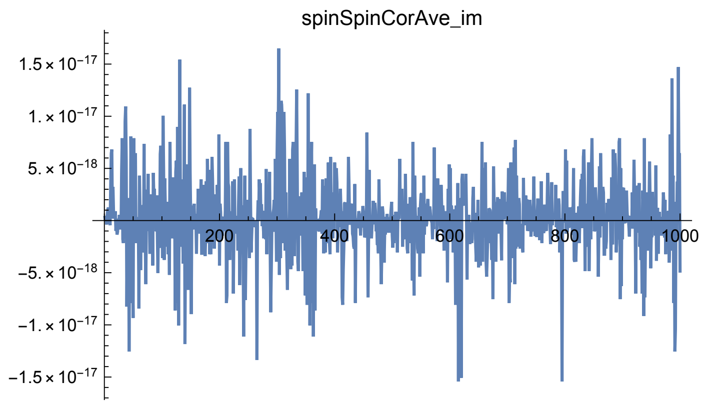
Out[450]=



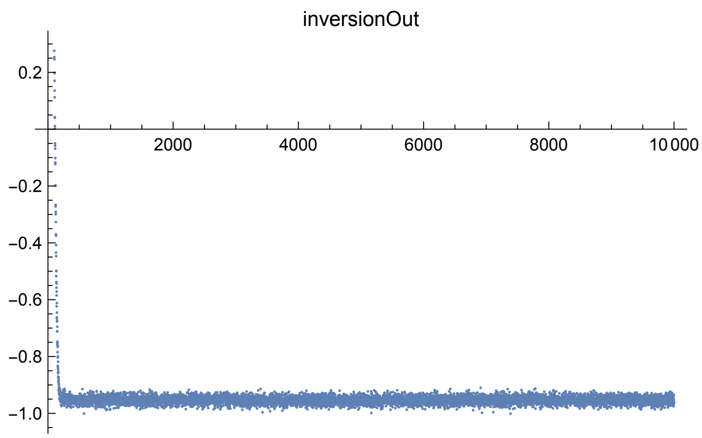
Out[451]=



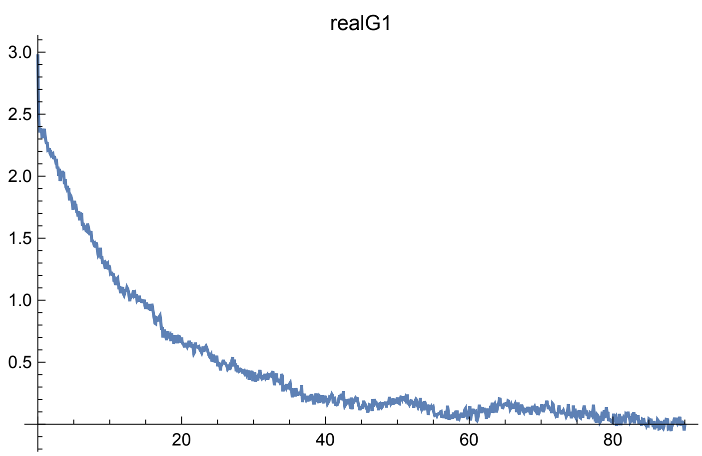
Out[452]=



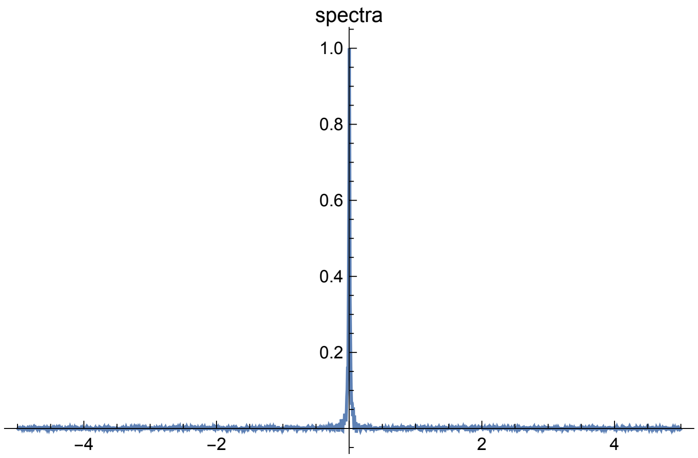
Out[453]=



Out[454]=



Out[455]=



In[456]:=
```
(*Fit the spectra to lorentzian*)
lorentzianModel = A linewidth/(linewidth^2 + 4 x^2);
fitLoren = NonlinearModelFit[spectra, {lorentzianModel}, {linewidth, A}, {x}]
fitLoren["ParameterTable"]
linewidth1 = linewidth /. fitLoren["BestFitParameters"]
Show[
 {Plot[fitLoren[x], {x, -0.3, 0.3}, PlotRange → All, BaseStyle → {FontSize → 30}],
  Graphics[{Red, PointSize[.01], Map[Point, spectra]}]},
 Frame → True, FrameLabel → {"ν/Transit time^-1", "Power Spectra"}]
```

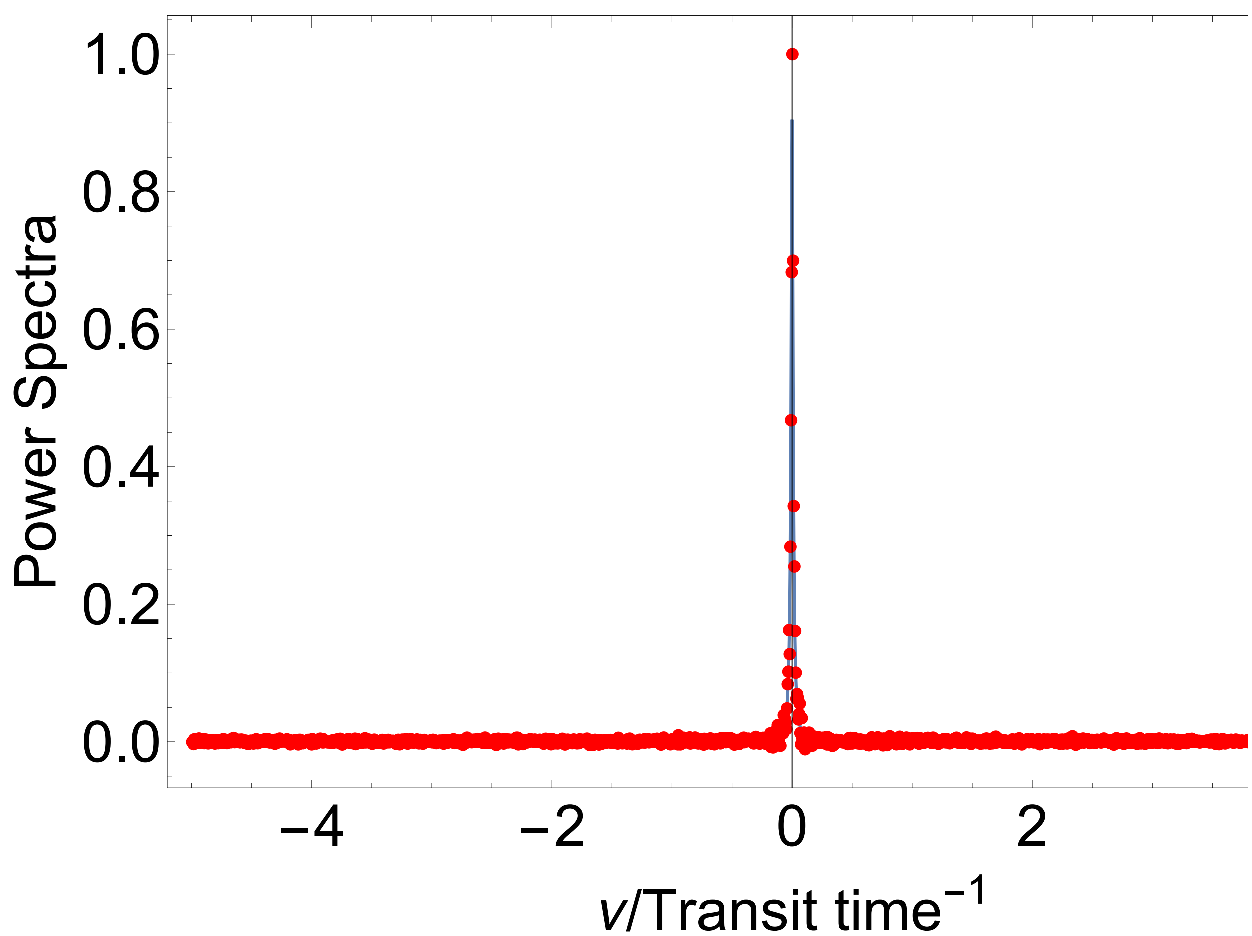
Out[457]= FittedModel[$\frac{0.000410944}{0.000454238+4x^2}$]

Out[458]=

	Estimate	Standard Error	t-Statistic	P-Value
linewidth	0.0213129	0.000209028	101.962	$8.19738839194\times10^{-750}$
A	0.0192815	0.000133264	144.686	$5.34965924147\times10^{-993}$

Out[459]= 0.0213129

Out[460]=



In[461]:=
```
tauc1 = 1/(linewidth1 Pi)
```
$$\text{tauc1} = \frac{1}{\text{linewidth1 Pi}}$$

Out[461]= 14.9351

In[462]:=
```
(*Fit the G1 function to exponential*)
exponentialModel = B Exp[- t / tc]
fitExp = NonlinearModelFit[realG1, {exponentialModel}, {tc, B}, {t}]
fitExp["ParameterTable"]
tauc2 = tc /. fitExp["BestFitParameters"]
Show[{Plot[fitExp[x], {x, 0, Last[realG1][[1]]}, PlotRange → All,
    PlotStyle → {Thickness[0.01]}, BaseStyle → {FontSize → 30}],
   Graphics[{Red, PointSize[.001], Map[Point, realG1]}]},
  Frame → True, FrameLabel → {"Time/T", "G1Slow"}]
```

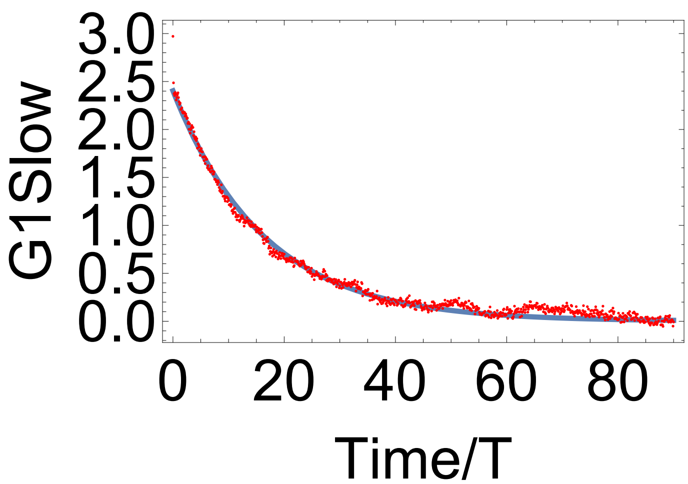
Out[462]= $B\, e^{-\frac{t}{tc}}$

Out[463]= FittedModel[$2.40327\, e^{-0.0611085\, t}$]

Out[464]=

	Estimate	Standard Error	t-Statistic	P-Value
tc	16.3643	0.0878847	186.202	$2.700743553834 \times 10^{-720}$
B	2.40327	0.00909324	264.292	$1.558193007379 \times 10^{-854}$

Out[465]= 16.3643

Out[466]=


In[467]:=
```
(*Get the linewidth from the coherent time*)
linewidth2 = 1/(tauc2 Pi)
```
$$\text{linewidth2} = \frac{1}{\text{tauc2 Pi}}$$

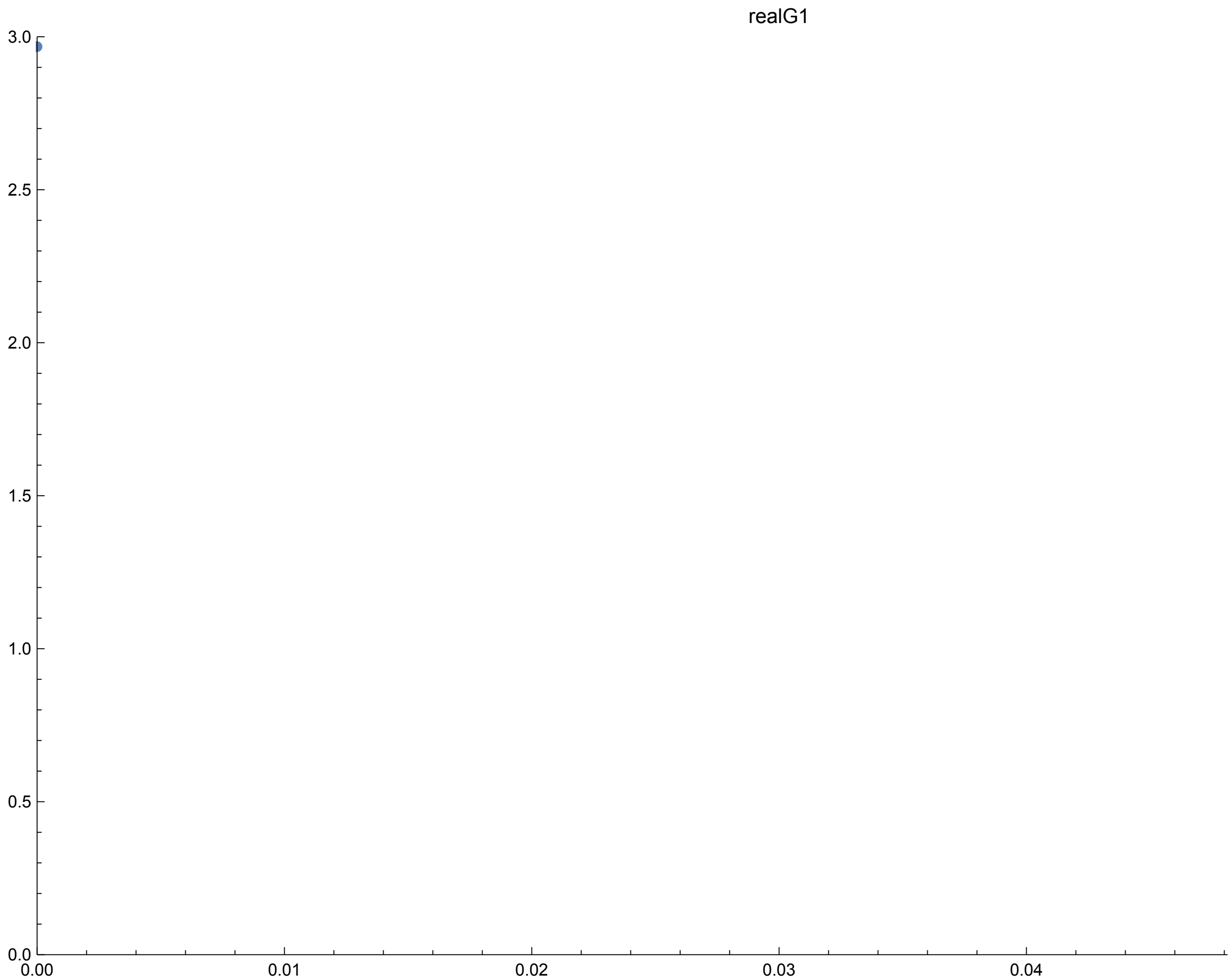
Out[467]= 0.0194514

In[468]:=

In[469]:=
```
(*Notice there is another exponential decay.*)
```

In[470]:=
```
cutOff = 0.06;
ListPlot[realG1, PlotRange → {{0, cutOff}, {0, 3}},
 PlotLabel → "realG1", PlotMarkers → {Automatic, Small}]
secondNum = IntegerPart[cutOff / realG1[[2, 1]]];
secondExp = Take[realG1, secondNum];
exponentialModel2 = B2 Exp[- Pi linewidthQuick t]
fitExp2 =
 NonlinearModelFit[secondExp, {exponentialModel2}, {linewidthQuick, B2}, {t}]
fitExp2["ParameterTable"]
Show[{Plot[fitExp2[x], {x, 0, cutOff}, PlotRange → All, BaseStyle → {FontSize → 30}],
  Graphics[{Red, PointSize[.02], Map[Point, secondExp]}]},
 Frame → True, FrameLabel → {"Time/T", "G1Fast"}]
```

Out[471]=



Out[474]= $B2\, e^{-\text{linewidthQuick}\, \pi\, t}$

NonlinearModelFit: First argument {} in NonlinearModelFit is not a list or a rectangular array.

Out[475]= NonlinearModelFit[{}, {$B2\, e^{-\text{linewidthQuick}\, \pi\, t}$}, {linewidthQuick, B2}, {t}]

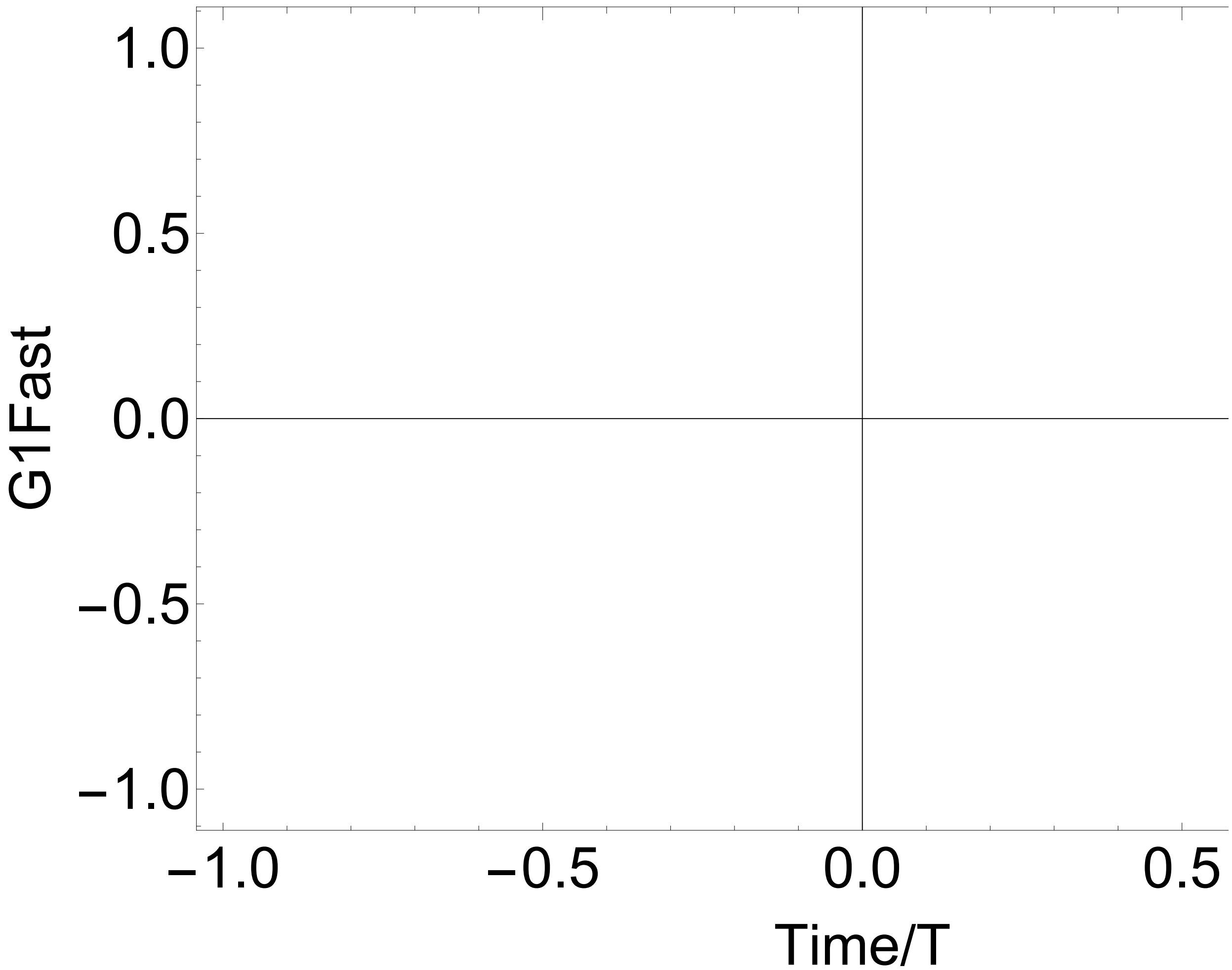
Out[476]= `NonlinearModelFit[{}, {B2 e^(-linewidthQuick π t)}, {linewidthQuick, B2}, {t}][ParameterTable]`

$$\text{NonlinearModelFit}\left[\{\},\ \left\{B2\ e^{-\text{linewidthQuick}\ \pi\ t}\right\},\ \{\text{linewidthQuick},\ B2\},\ \{t\}\right][\text{ParameterTable}]$$

••• NonlinearModelFit: First argument {} in NonlinearModelFit is not a list or a rectangular array.

••• NonlinearModelFit: First argument {} in NonlinearModelFit is not a list or a rectangular array.

Out[477]=



In[478]:=

$$\text{secondTc} = \frac{1}{\text{linewidthQuick Pi}}\ \text{/. fitExp2["BestFitParameters"]}$$

••• ReplaceAll: $\left\{\text{NonlinearModelFit}\left[\{\},\ \left\{B2\ e^{-\text{linewidthQuick}\ \pi\ t}\right\},\ \{\text{linewidthQuick},\ B2\},\ \{t\}\right][\text{BestFitParameters}]\right\}$ is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing.

Out[478]=

$$\frac{1}{\text{linewidthQuick}\ \pi}\ \text{/.}\ \text{NonlinearModelFit}\left[\{\},\ \left\{B2\ e^{-\text{linewidthQuick}\ \pi\ t}\right\},\ \{\text{linewidthQuick},\ B2\},\ \{t\}\right][\text{BestFitParameters}]$$