Sin(2β) fit test:

The units of Γ, ΔΓ, and Δm are ps-1

In[765]:=
```
Γ = 1.530;
Δm = 0.507;
ΔΓ = 0;
```

PDF for the Sin(2β) analysis:

Model parameters:

In[768]:=
```
Norma = 1;
Sf = 0.672;
Cf = 0;
Df = 1;
```

Full Model with unitarity considerations:

In[772]:=
```
f[t_] := Exp[-Γ Abs[t]] Norma (Cosh[ΔΓ t] + Df Sinh[ΔΓ t] + Sf Sin[Δmd t] + Cf Cos[Δmd t])
```

Model without any violation effect:

In[773]:=
```
f0[t_] := Exp[-Γ Abs[t]] Norma (Cosh[ΔΓ t])
```

MonteCarlo generation:

- **Half sized Sample (Δt>0):**

In[774]:=
```
nbins = 50;
x0 = 0;
xf = 10;
y0 = 0;
yf = 1;
paso = (xf - x0) / nbins;
pos = x0 + paso / 2;
Npoints = 10 000;
```

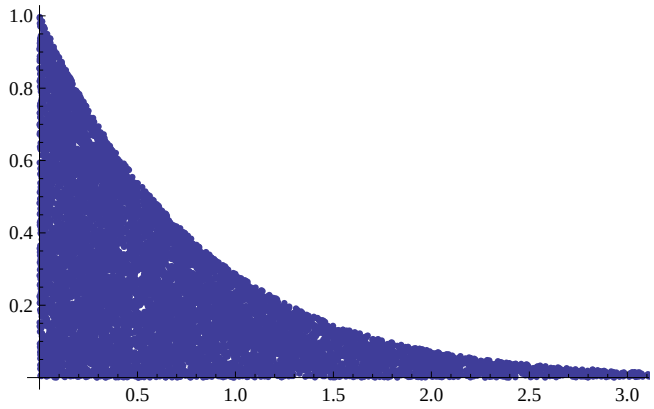
In[782]:=
```
Lista = {};
ListHistogram = {};
DataList = {};
```

In[785]:=
```
i = 0;
```

In[786]:=
```
While[i < Npoints,
 x = RandomReal[{x0, xf}]; y = RandomReal[{y0, yf}];
 If[y < f[x],
  AppendTo[Lista, {x, y}];
  AppendTo[ListHistogram, x];
  i++]
]
CountList = BinCounts[ListHistogram, {x0, xf, paso}];
```

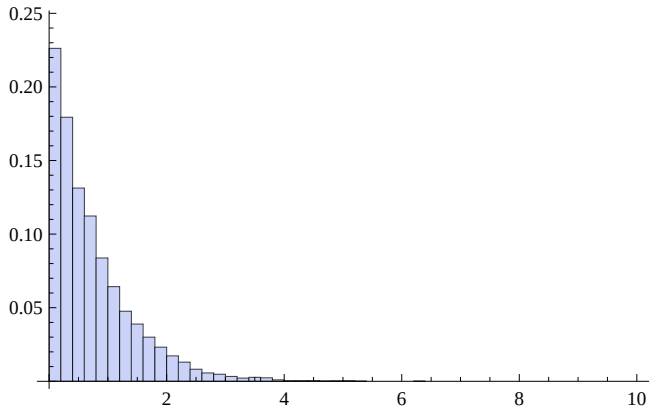
In[788]:=
```
ListPlot[Lista]
```

Out[788]=



In[789]:=
```
Hpos = Histogram[ListHistogram, {x0, xf, paso}, "Probability"]
```

Out[789]=



- **Fitting the data:**

In[790]:=
```
For[i = 1, i < nbins, i++,
      AppendTo[DataList, {pos, CountList[[i]] / Npoints}];
      pos += paso;
]
```

In[791]:=
```
DataList = N[DataList];
```

In[792]:=
```
model = Exp[- Γ Abs[t]] n (1 + c Cos[Δm t] + s Sin[Δm t]);
```

In[793]:=
```
fit1 = FindFit[DataList, model, {{n, DataList[[1, 2]]}, {s, 0.672}, {c, 0.}}, t]
```

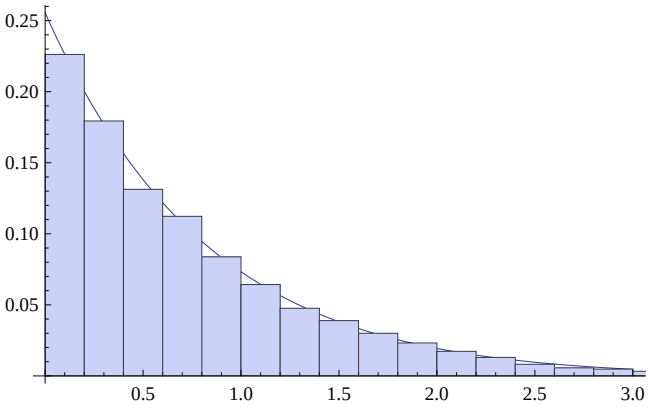
Out[793]=
```
{n → 0.304238, s → 0.519415, c → -0.15999}
```

In[794]:=
```
(*fit1=FindFit[DataList,f[t],{{Norma,DataList[[1,2]]},{Sf,0.7},{Cf,0.1}},t]*)
```

Show the histogram with its fit.

In[795]:=
```
Show[Plot[model /. fit1, {t, 0, 3}], Hpos]
```

Out[795]=



■ Half sized Sample (Δt<0):

In[796]:=
```
nbins = 50;
x0 = -10;
xf = 0;
y0 = 0;
yf = 1;
paso = (xf - x0) / nbins;
pos = x0 + paso / 2;
Npoints = 10 000;
```

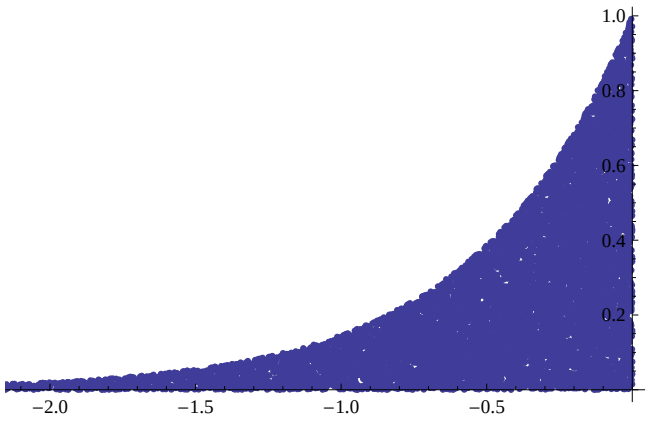
In[804]:=
```
Lista = {};
ListHistogram = {};
DataList = {};
```

In[807]:=
```
i = 0;
```

In[808]:=
```
While[i < Npoints,
 x = RandomReal[{x0, xf}]; y = RandomReal[{y0, yf}];
 If[y < f[x],
  AppendTo[Lista, {x, y}];
  AppendTo[ListHistogram, x];
  i++]
]
CountList = BinCounts[ListHistogram, {x0, xf, paso}];
```

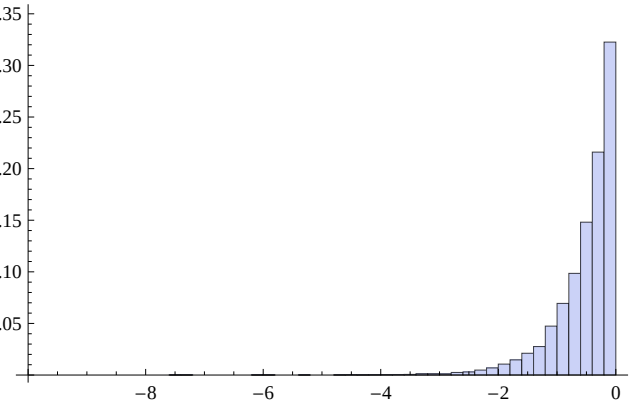
In[810]:=
```
ListPlot[Lista]
```

Out[810]=



In[811]:=
```
Hneg = Histogram[ListHistogram, {x0, xf, paso}, "Probability"]
```

Out[811]=



- **Fitting the data:**

In[812]:=
```
For[i = 1, i < nbins, i++,
      AppendTo[DataList, {pos, CountList[[i]] / Npoints}];
      pos += paso;
]
```

In[813]:=
```
DataList = N[DataList];
```

In[814]:=
```
model2 = Exp[- Γ Abs[t]] n (1 + c Cos[Δm t] + s Sin[Δm t]);
```

In[815]:=
```
fit2 = FindFit[DataList, model, {{n, DataList[[1, 2]]}, {s, -0.672}, {c, 0.}}, t]
```

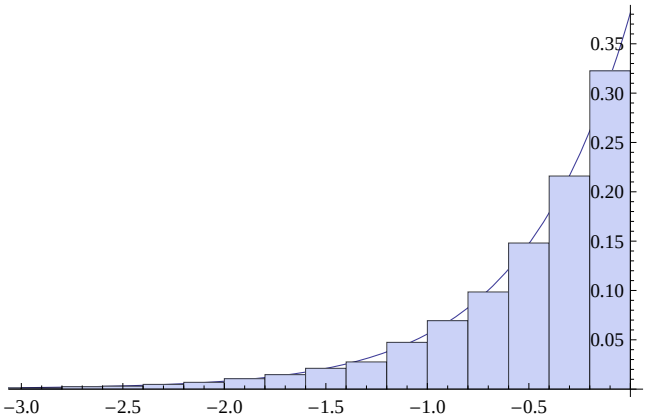
Out[815]=
```
{n → 0.407017, s → 0.642914, c → -0.0627692}
```

In[816]:=
```
(*fit1=FindFit[DataList,f[t],{{Norma,DataList[[1,2]]},{Sf,0.7},{Cf,0.1}},t]*)
```

Show the histogram with its fit.

In[817]:=
```
Show[Plot[model2 /. fit2, {t, -3, 0}], Hneg]
```

Out[817]=



■ Full sample (Δt>0 && Δt<0):

In[818]:=
```
nbins = 100;
x0 = -10;
xf = 10;
y0 = 0;
yf = 1;
paso = (xf - x0) / nbins;
pos = x0 + paso / 2;
Npoints = 10 000;
```

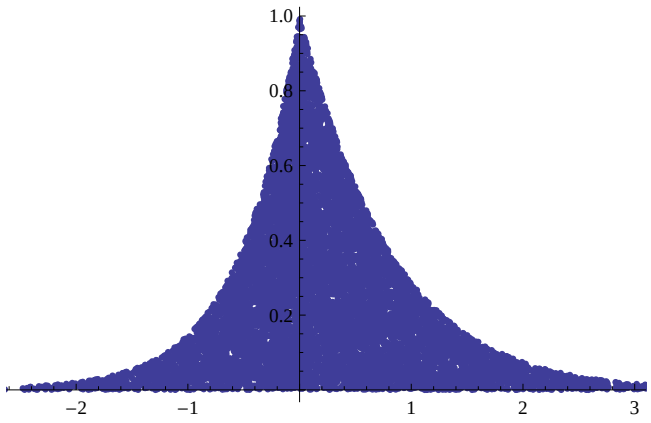
In[826]:=
```
Lista = {};
ListHistogram = {};
DataList = {};
```

In[829]:=
```
i = 0;
```

In[830]:=
```
While[i < Npoints,
 x = RandomReal[{x0, xf}]; y = RandomReal[{y0, yf}];
 If[y < f[x],
  AppendTo[Lista, {x, y}];
  AppendTo[ListHistogram, x];
  i++]
]
CountList = BinCounts[ListHistogram, {x0, xf, paso}];
```

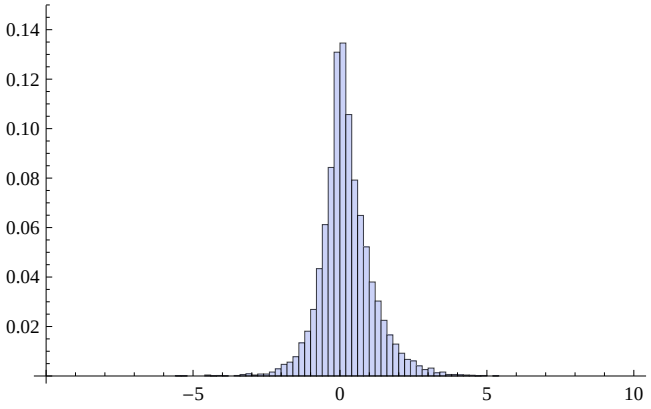
In[832]:=
```
ListPlot[Lista]
```

Out[832]=



In[833]:=
```
H = Histogram[ListHistogram, {x0, xf, paso}, "Probability"]
```

Out[833]=



- **Fitting the data:**

In[834]:=
```
For[i = 1, i < nbins, i++,
      AppendTo[DataList, {pos, CountList[[i]] / Npoints}];
      pos += paso;
]
```

In[835]:=
```
DataList = N[DataList];
```

In[836]:=
```
model2 = Exp[- Γ Abs[t]] n (1 + c Cos[Δm t] + s Sin[Δm t]);
```

In[841]:=
```
fit2 = FindFit[DataList, model, {n, {s, -0.672}, {c, 0.}}, t]
```

Out[841]=
```
{n → 0.158414, s → 0.610688, c → -0.0328191}
```

Show the histogram with its fit.

In[842]:=
```
Show[Plot[model2 /. fit2, {t, -3, 3}], H]
```

Out[842]=