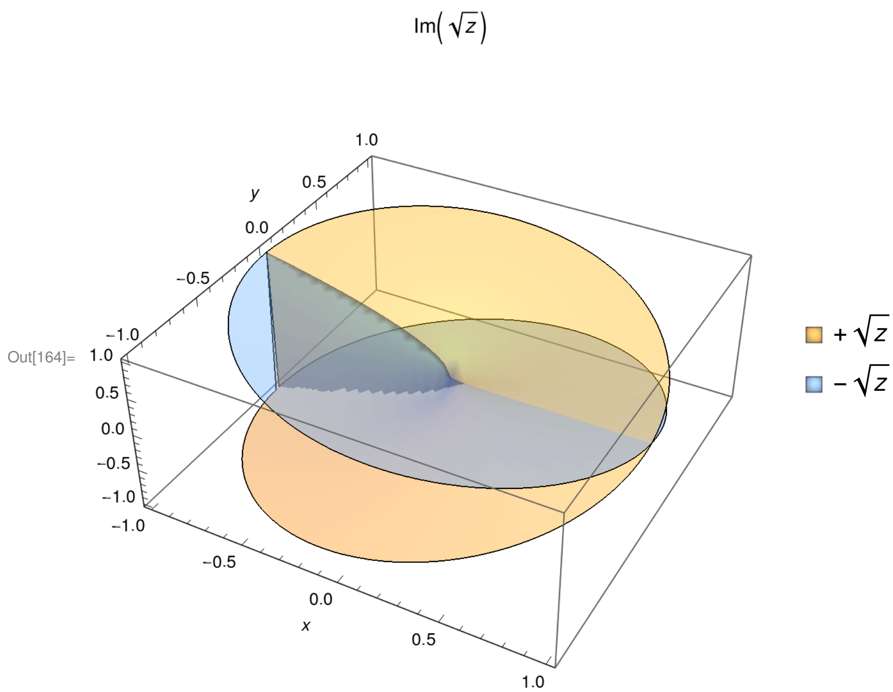
Example of $\sqrt{z}$ Riemann sheets

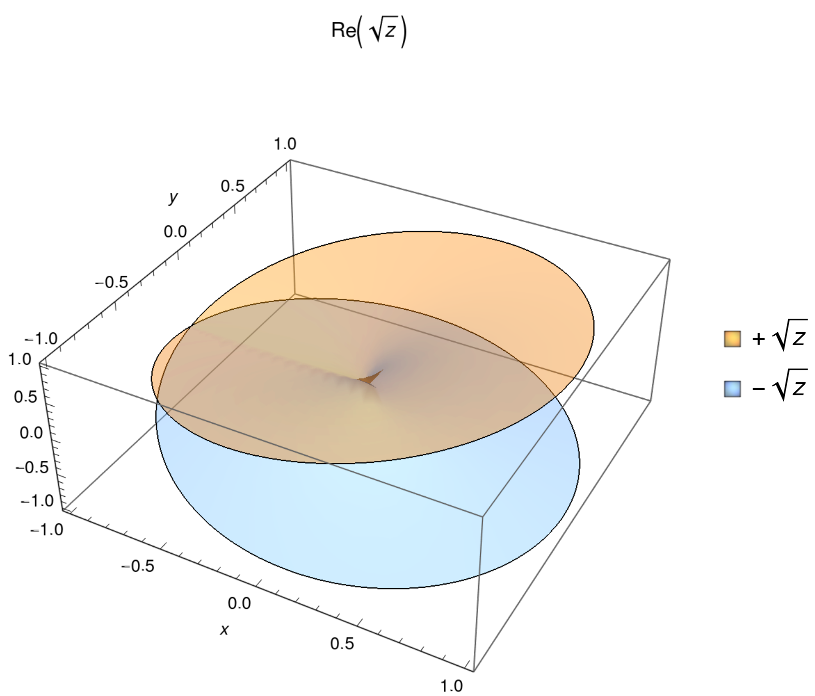
These are the definitions by Mathematica, which we don't like very much...

In[164]:=
```
Plot3D[{Im[√(x + ⅈ y)], Im[-√(x + ⅈ y)]}, {x, -1, 1}, {y, -1, 1},
 RegionFunction → Function[{x, y, z}, 1 ≥ x^2 + y^2], ExclusionsStyle → Opacity[1],
 Mesh → False, PlotStyle → Opacity[0.4], AxesLabel → Automatic,
 PlotLegends → {"+ √z ", "- √z "}, PlotLabel → "Im( √z )", ImageSize → Medium]
Plot3D[{Re[√(x + ⅈ y)], Re[-√(x + ⅈ y)]}, {x, -1, 1},
 {y, -1, 1}, RegionFunction → Function[{x, y, z}, 1 ≥ x^2 + y^2],
 Mesh → False, PlotStyle → Opacity[0.4], AxesLabel → Automatic,
 PlotLegends → {"+ √z ", "- √z "}, PlotLabel → "Re( √z )", ImageSize → Medium]
```

Out[164]=



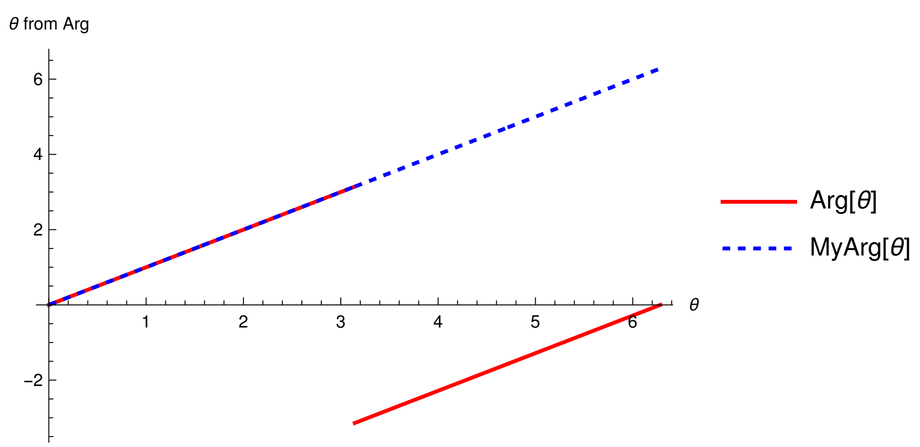
Out[165]=



Here I define our appropriate versions of $\sqrt{z}$. For that, I need the MyArg[z] function

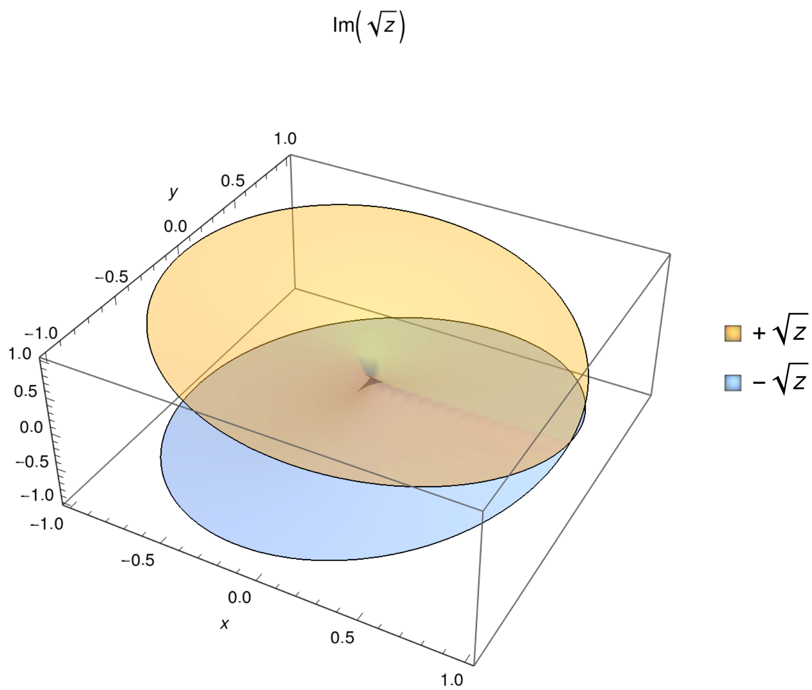
In[166]:=
```
MyArg[z_] := Arg[z] + If[Im[z] < 0, 2 π, 0];
Plot[{Arg[Exp[ⅈ θ]], MyArg[Exp[ⅈ θ]]}, {θ, 0, 2 π},
 PlotStyle → {{Red, Thick}, {Blue, Thick, Dashed}},
 AxesLabel → {"θ", "θ from Arg"}, PlotLegends → {"Arg[θ]", "MyArg[θ]"}]
```

Out[167]=

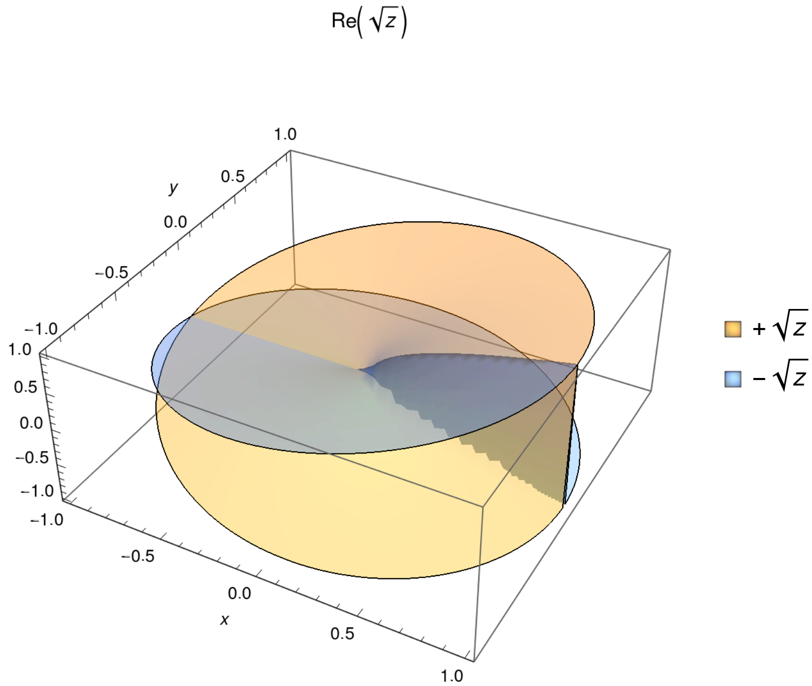


In[168]:=
```
MySqrt[z_] := Abs[√z] Exp[1/2 ⅈ MyArg[z]];
Plot3D[{Im[MySqrt[x + ⅈ y]], Im[-MySqrt[x + ⅈ y]]}, {x, -1, 1}, {y, -1, 1},
 RegionFunction → Function[{x, y, z}, 1 ≥ x^2 + y^2], ExclusionsStyle → Opacity[1],
 Mesh → False, PlotStyle → Opacity[0.4], AxesLabel → Automatic,
 PlotLegends → {"+ √z ", "- √z "}, PlotLabel → "Im(√z)", ImageSize → Medium]
Plot3D[{Re[MySqrt[x + ⅈ y]], Re[-MySqrt[x + ⅈ y]]}, {x, -1, 1},
 {y, -1, 1}, RegionFunction → Function[{x, y, z}, 1 ≥ x^2 + y^2],
 Mesh → False, PlotStyle → Opacity[0.4], AxesLabel → Automatic,
 PlotLegends → {"+ √z ", "- √z "}, PlotLabel → "Re(√z)", ImageSize → Medium]
```

Out[169]=



Out[170]=



Definition of the $G_{\mathrm{I}}(s)$ function, and IIRS $G_{\mathrm{II}}(s)$

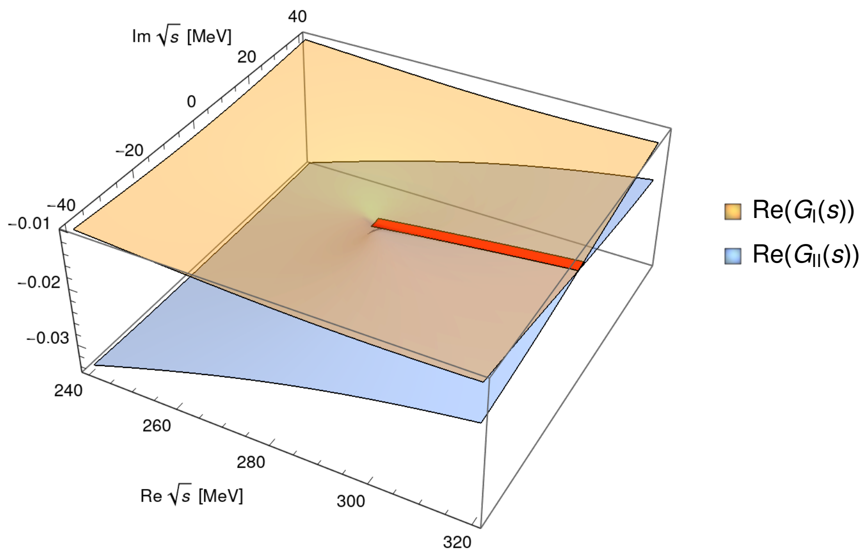
This can be applied to our definition of the $G(s)$ function.

In[171]:=

```
σ[s_] := Abs[Sqrt[1 - (4 m^2)/s]] Exp[(ⅈ/2) MyArg[1 - (4 m^2)/s]];
G[s_] := 1/(16 π^2) (g0 - σ[s] Log[(σ[s] - 1)/(σ[s] + 1)]);
GII[s_] := G[s] + 2 ⅈ σ[s]/(16 π);
NumValsE = {fπ → 87.3, GV → Sqrt[(gv^2 fπ^2)/2], gv → 1, m → 139, Mρ → 770, g0 → -3.0};
pImG = Plot3D[{Im[G[(Wr + ⅈ Wi)^2] //. NumValsE], Im[GII[(Wr + ⅈ Wi)^2] //. NumValsE]},
   {Wr, 240, 320}, {Wi, -40, 40}, Mesh → False, PlotStyle → Opacity[0.4],
   AxesLabel → {"Re √s  [MeV]", "Im √s  [MeV]"}, PlotPoints → 60,
   PlotLabel → "Im(G(s))", PlotLegends → {"Im(G_I(s))", "Im(G_II(s))"}];
pReG = Plot3D[{Re[G[(Wr + ⅈ Wi)^2] //. NumValsE], Re[GII[(Wr + ⅈ Wi)^2] //. NumValsE]},
   {Wr, 240, 320}, {Wi, -40, 40}, Mesh → False, PlotStyle → Opacity[0.4],
   AxesLabel → {"Re √s  [MeV]", "Im √s  [MeV]"}, PlotPoints → 60,
   PlotLabel → "Re(G(s))", PlotLegends → {"Re(G_I(s))", "Re(G_II(s))"}];
pReGCut = Plot3D[{Re[G[Wr^2 + 0.0001 ⅈ] //. NumValsE]},
   {Wr, 2 m //. NumValsE, 320}, {Wi, -40, 40}, PlotPoints → 60, Mesh → False,
   PlotStyle → {Red}, AxesLabel → {"Re √s  [MeV]", "Im √s  [MeV]"},
   RegionFunction → Function[{Wr, Wi, z}, (0 ≤ Wi ≤ 3) && (Wr > 2 m //. NumValsE)]];
pImGCutTop = Plot3D[Im[G[Wr^2 + 0.0001 ⅈ] //. NumValsE],
   {Wr, 2 m //. NumValsE, 320}, {Wi, -40, 40}, PlotPoints → 60, Mesh → False,
   PlotStyle → {Red}, AxesLabel → {"Re √s  [MeV]", "Im √s  [MeV]"},
   RegionFunction → Function[{Wr, Wi, z}, (0 ≤ Wi ≤ 3) && (Wr > 2 m //. NumValsE)]];
pImGCutBot = Plot3D[Im[G[Wr^2 - 0.0001 ⅈ] //. NumValsE],
   {Wr, 2 m //. NumValsE, 320}, {Wi, -40, 40}, PlotPoints → 60, Mesh → False,
   PlotStyle → {Blue}, AxesLabel → {"Re √s  [MeV]", "Im √s  [MeV]"},
   RegionFunction → Function[{Wr, Wi, z}, (-3 ≤ Wi ≤ 0) && (Wr > 2 m //. NumValsE)]];
Show[pReG, pReGCut]
Show[pImG, pImGCutTop, pImGCutBot]
```

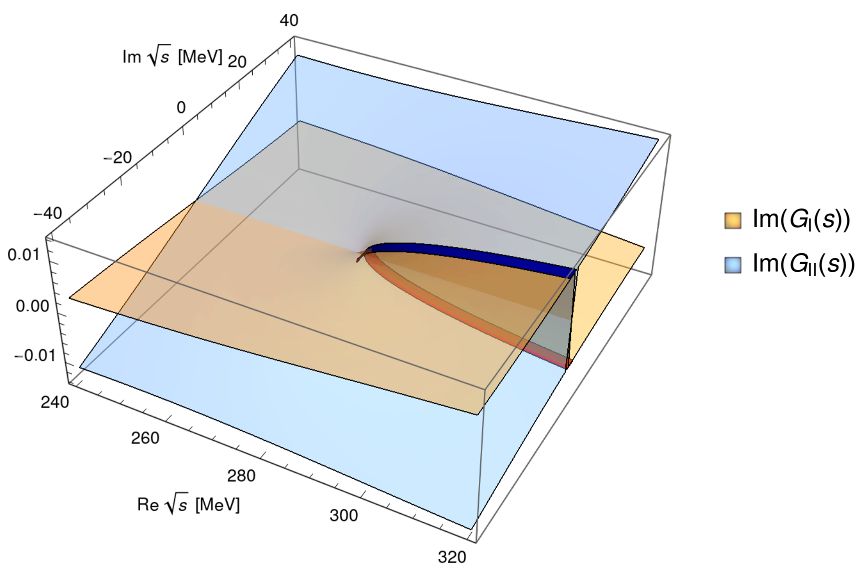
Re(G(s))

Out[180]=



Im(G(s))

Out[181]=



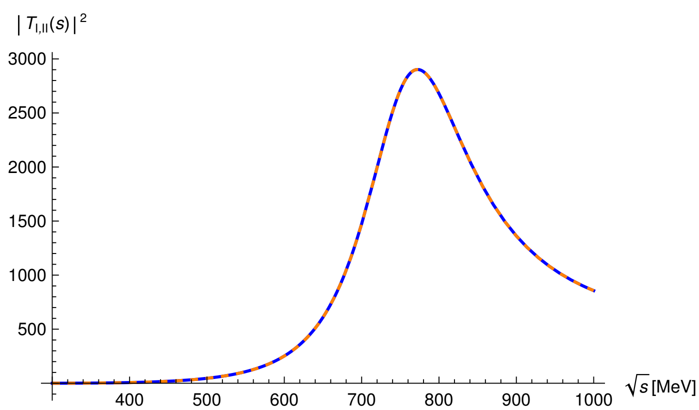
Definition of the *T*-matrix *T*(*s*)

In[182]:=
```
h[a_] := -2/3 (1 + 6 a + 3 a^2) + a (2 + 3 a + a^2) Log[1 + 2/a];
p2[s_] := s/4 - m^2;
V1[s_] := -(2 p2[s])/(3 fπ^2) (1 - GV^2/fπ^2 (2 s)/(s - Mρ^2)) - GV^2/fπ^4 p2[s] × h[Mρ^2/(2 p2[s])];
TI[s_] := 1/(1/V1[s] - G[s]);
TII[s_] := 1/(1/V1[s] - GII[s]);
SMat[s_] := 1 - 2 ⅈ Sqrt[p2[s]]/(8 π Sqrt[s]) TI[s];
δ[s_] := 1/2 MyArg[SMat[s]];
```

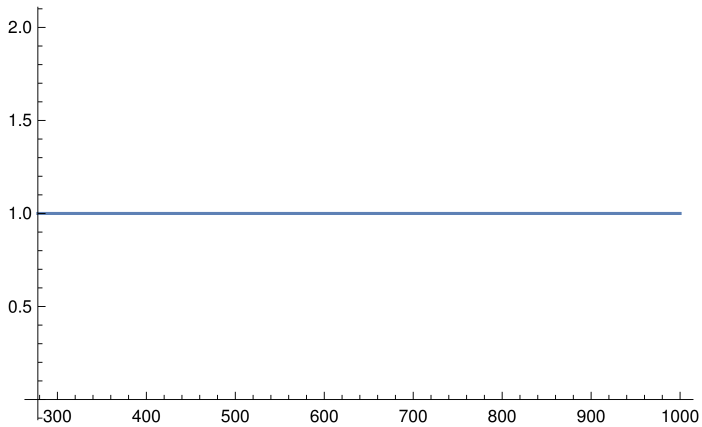
Some plots for *T*(*s*), *S*(*s*) and *δ*(*s*)

In[189]:=
```
Plot[{Abs[TI[W^2]]^2 //. NumValsE, Abs[TII[W^2]]^2 //. NumValsE}, {W, 300, 1000},
 PlotStyle → {Blue, {Orange, Dashed}}, AxesLabel → {"√s [MeV]", "|T_I,II(s)|^2"}]
Plot[{Abs[SMat[W^2] //. NumValsE]}, {W, 2 m //. NumValsE, 1000}]
(SMat[W^2] - 1)/(2 ⅈ) //. NumValsE //. {W → 500.}
ParametricPlot[{Re[SMat[W^2]] //. NumValsE, Im[SMat[W^2]] //. NumValsE},
 {W, 2 m //. NumValsE, 1200}, PlotRange → {-1.2, 1.2}, AxesLabel → {"Re[S(s)]", "Im[S(s)]"}]
```

Out[189]=

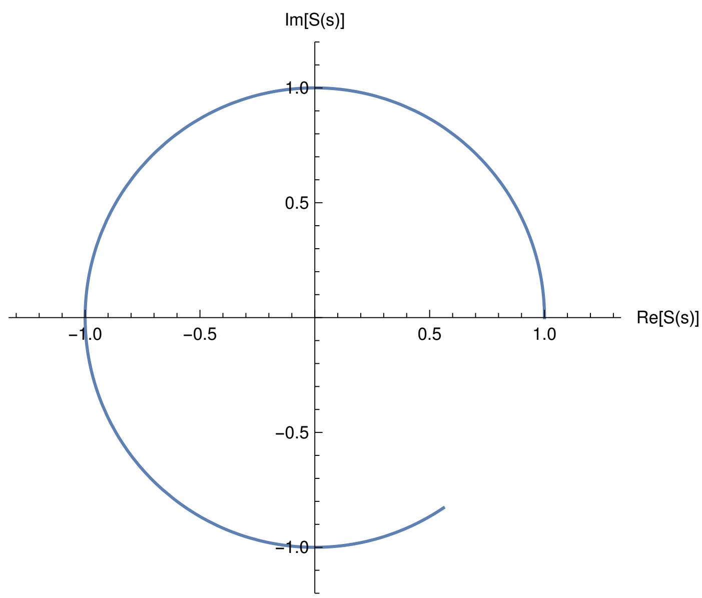


Out[190]=



Out[191]= 0.111113 + 0.0125024 ⅈ

Out[192]=

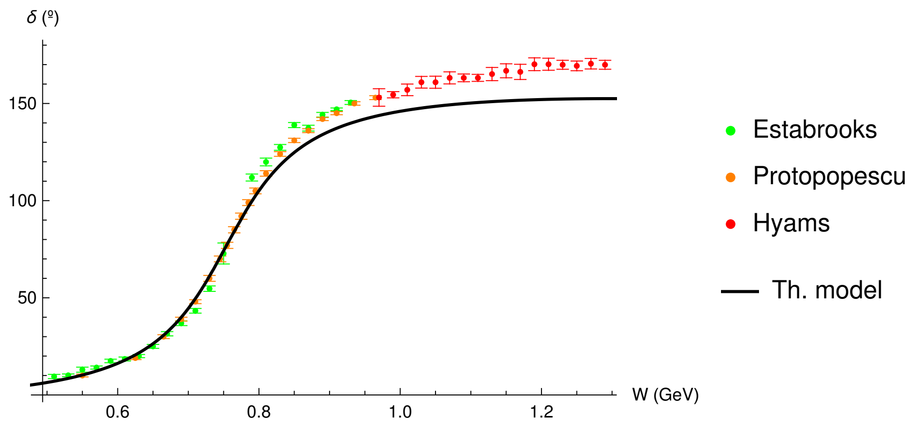


Import the data for the phase shifts

In[193]:=
```
EstaRaw =
  Import["/home/albala/Dropbox/Projects/2023/SalamancaSchool/PiPi_I1/estabrooks.dat"];
NDatEsta = 21;
ProtoRaw = Import[
   "/home/albala/Dropbox/Projects/2023/SalamancaSchool/PiPi_I1/protopopescu.dat"];
NDatProto = 20;
HyamsRaw = Import[
   "/home/albala/Dropbox/Projects/2023/SalamancaSchool/PiPi_I1/inelasticphasehyams73.
     dat"];
NDatHyams = 17;
```

In[257]:=
```
DataPlot =
  ListPlot[{Table[{EstaRaw[[i, 1]], Around[EstaRaw[[i, 2]], EstaRaw[[i, 4]]]}, {i, 1, NDatEsta}],
    Table[{ProtoRaw[[i, 1]], Around[ProtoRaw[[i, 2]], ProtoRaw[[i, 4]]]}, {i, 1, NDatProto}],
    Table[{HyamsRaw[[i, 1]], Around[HyamsRaw[[i, 2]], HyamsRaw[[i, 3]]]}, {i, 1, NDatHyams}]},
   PlotStyle → {{Green, PointSize[0.01]}, {Orange, PointSize[0.01]},
     {Red, PointSize[0.01]}}, PlotLegends → {"Estabrooks", "Protopopescu", "Hyams"}];
TheoPlot = Plot[180/π δ[(1000 W)^2] //. NumValsE, {W, (2 m/1000 //. NumValsE), 1.4},
   PlotStyle → {Black}, PlotLegends → {"Th. model"}];
Show[DataPlot, TheoPlot, AxesLabel → {"W (GeV)", "δ (°)"}]
```

Out[258]=



In[201]:=
```
FixedValuesFit = {m → 139, fπ → 87, Mρ → 750.001};
EstaDataFit = Table[{EstaRaw[[i, 1]], EstaRaw[[i, 2]], 3 EstaRaw[[i, 4]]}, {i, 1, NDatEsta}];
ProtoDataFit =
  Table[{ProtoRaw[[i, 1]], ProtoRaw[[i, 2]], 3 ProtoRaw[[i, 4]]}, {i, 1, NDatProto}];
HyamsDataFit = Table[{HyamsRaw[[i, 1]], HyamsRaw[[i, 2]], HyamsRaw[[i, 3]]}, {i, 1, NDatHyams}];
DataFit = Join[EstaDataFit, ProtoDataFit, HyamsDataFit];
fit = NonlinearModelFit[DataFit[[All, {1, 2}]],
   180/π δ[(1000 x)^2] //. FixedValuesFit, {{g0, -3}, {GV, 60}}, x,
   Weights → 1 / DataFit[[All, 3]]^2, VarianceEstimatorFunction → (1 &)];
fit["ParameterTable"]
fit["BestFitParameters"]
fit["ANOVATableSumsOfSquares"][[2]] / (Length[DataFit] - 2)
MatrixForm[fit["CorrelationMatrix"]]
```

Out[207]=

	Estimate	Standard Error	t-Statistic	P-Value
g0	-2.11074	0.0695845	-30.3335	1.91175×10^{-36}
GV	55.1916	0.268658	205.434	2.8163×10^{-82}

Out[208]= {g0 → -2.11074, GV → 55.1916}

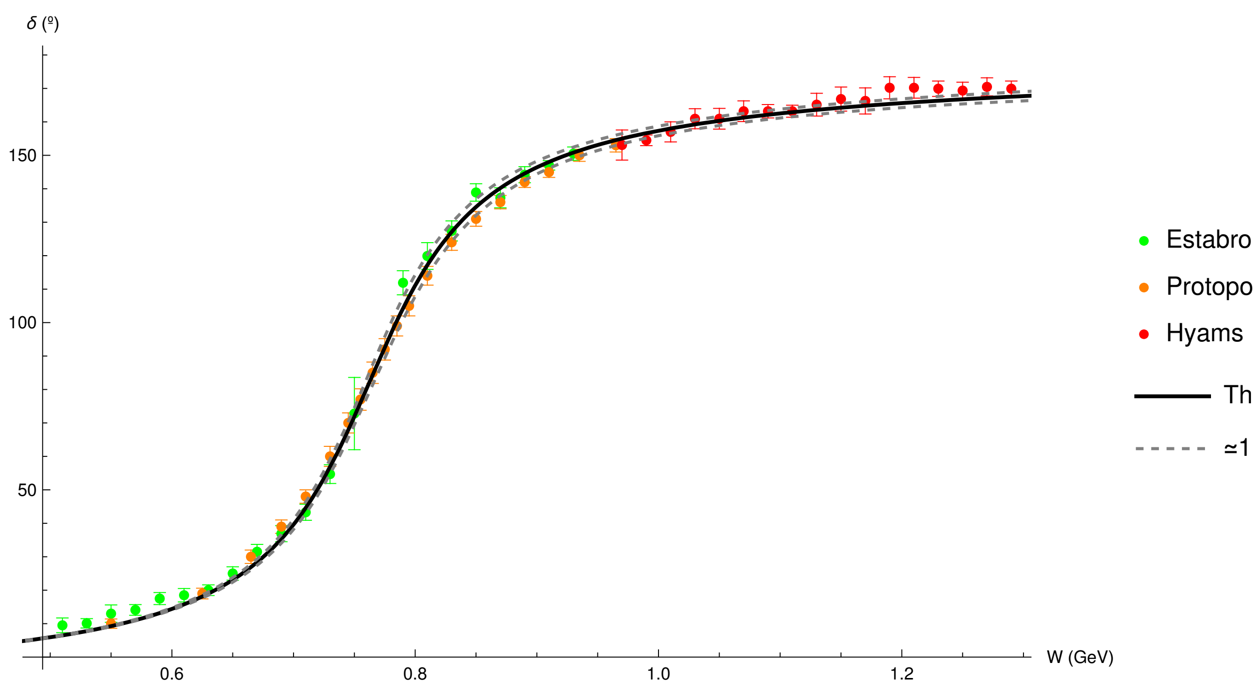
Out[209]= 0.741918

Out[210]//MatrixForm=

$$\begin{pmatrix} 1. & -0.461931 \\ -0.461931 & 1. \end{pmatrix}$$

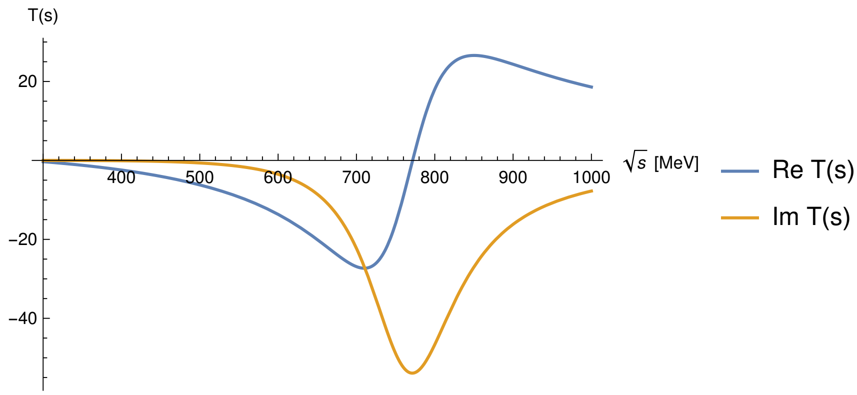
In[211]:=
```
g0min = fit["ParameterConfidenceIntervals"][[1, 1]];
g0max = fit["ParameterConfidenceIntervals"][[1, 2]];
GVmin = fit["ParameterConfidenceIntervals"][[2, 1]];
GVmax = fit["ParameterConfidenceIntervals"][[2, 2]];
SquaresParameters = {{g0 → g0min, GV → GVmin},
   {g0 → g0min, GV → GVmax}, {g0 → g0max, GV → GVmin}, {g0 → g0max, GV → GVmax}};
UpperValue[W_] :=
  Max[Table[180/π δ[(1000 W)^2] //. FixedValuesFit //. SquaresParameters[[i]], {i, 1, 4}]];
LowerValue[W_] :=
  Min[Table[180/π δ[(1000 W)^2] //. FixedValuesFit //. SquaresParameters[[i]], {i, 1, 4}]];
DataPlot =
 ListPlot[{Table[{EstaRaw[[i, 1]], Around[EstaRaw[[i, 2]], 2 EstaRaw[[i, 4]]]}, {i, 1, NDatEsta}],
   Table[{ProtoRaw[[i, 1]], Around[ProtoRaw[[i, 2]], 2 ProtoRaw[[i, 4]]]}, {i, 1, NDatProto}],
   Table[{HyamsRaw[[i, 1]], Around[HyamsRaw[[i, 2]], HyamsRaw[[i, 3]]]}, {i, 1, NDatHyams}]},
  PlotStyle → {{Green, PointSize[0.01]}, {Orange, PointSize[0.01]},
    {Red, PointSize[0.01]}}, PlotLegends → {"Estabrooks", "Protopopescu", "Hyams"}];
TheoPlot = Plot[{180/π δ[(1000 W)^2] //. FixedValuesFit //. fit["BestFitParameters"],
   UpperValue[W], LowerValue[W]}, {W, (2 m/1000 //. FixedValuesFit), 1.4},
  PlotStyle → {{Black, Thick}, {Gray, Dashed}, {Gray, Dashed}},
  PlotLegends → {"Th. model (fit)", "≃1σ bands"}, ImageSize → Large];
Show[{DataPlot, TheoPlot}, ImageSize → Large, AxesLabel → {"W (GeV)", "δ (°)"}]
```

Out[219]=

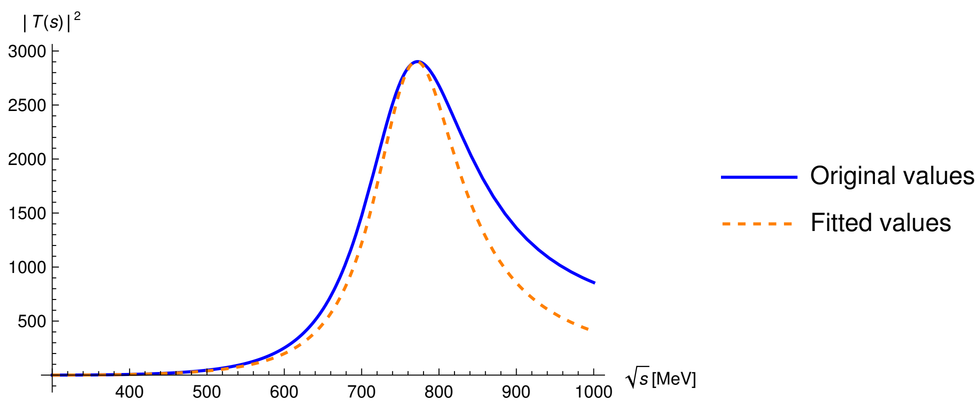


```
In[220]:= Plot[{Re[TI[W^2]] //. FixedValuesFit //. fit["BestFitParameters"]],
    Im[TI[W^2]] //. FixedValuesFit //. fit["BestFitParameters"]]}, {W, 300, 1000},
  PlotLegends → {"Re  T(s)", "Im  T(s)"}, AxesLabel → {" √s  [MeV]", "T(s)"}]
Plot[{Abs[TI[W^2]]^2 //. NumValsE,
    Abs[TII[W^2]]^2 //. FixedValuesFit //. fit["BestFitParameters"]}, {W, 300, 1000},
  PlotStyle → {Blue, {Orange, Dashed}}, AxesLabel → {" √s [MeV]", "|T(s)|^2"},
  PlotLegends → {"Original  values", "Fitted  values"}]
```

Out[220]=



Out[221]=

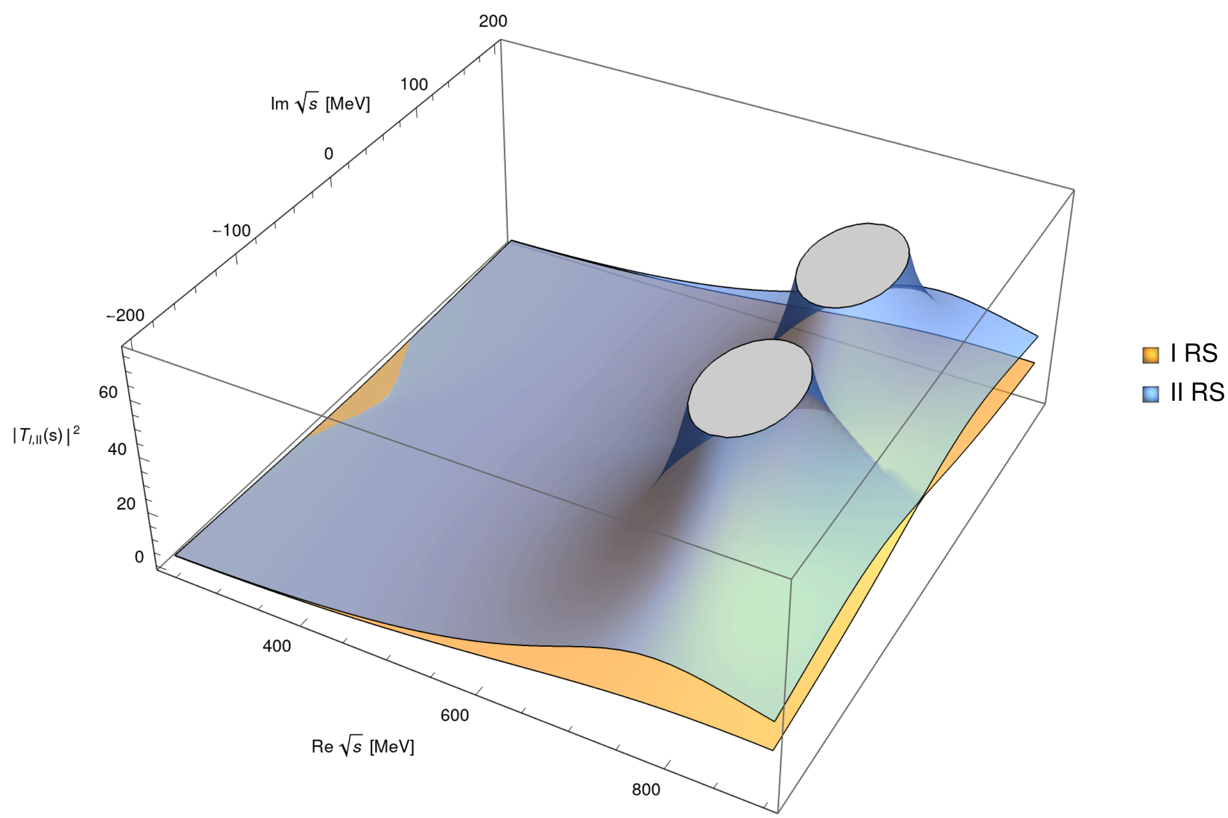


Finding the pole

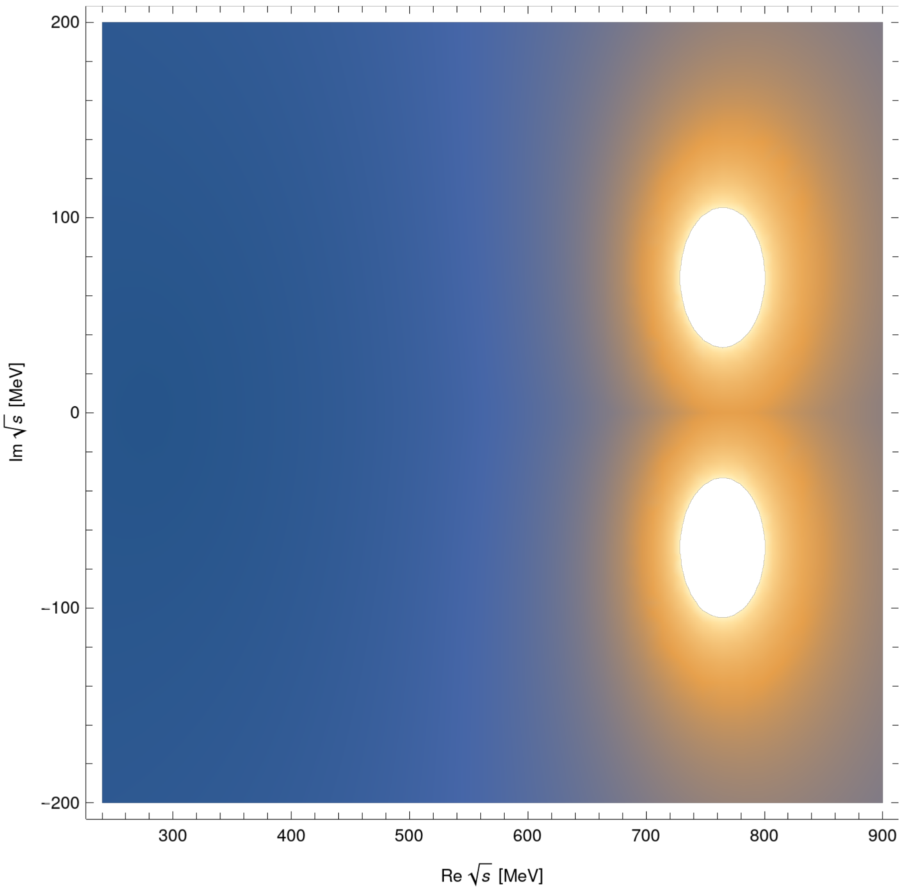
The previous plots of the amplitude show a prominent peak around $\sqrt{s}$ = 775 MeV. This is usually (but not always!) due to the presence of a pole in the II RS. We plot the amplitude in the whole complex plane for both sheets.

In[222]:=
```
pAbsT2ComplexPlane =
  Plot3D[{Abs[TI[(Wr + ⅈ Wi)^2]] //. FixedValuesFit //. fit["BestFitParameters"],
     Abs[TII[(Wr + ⅈ Wi)^2]] //. FixedValuesFit //. fit["BestFitParameters"]}, {Wr, 240, 900},
   {Wi, -200, 200}, PlotPoints → 40, Mesh → False, PlotStyle → Opacity[0.6],
   AxesLabel → {"Re √s  [MeV]", "Im √s  [MeV]", "|T_{I,II}(s)|^2"},
   PlotLegends → {"I RS", "II RS"}, ImageSize → Large]
DensityPlot[Abs[TII[(Wr + ⅈ Wi)^2]] //. FixedValuesFit //. fit["BestFitParameters"],
 {Wr, 240, 900}, {Wi, -200, 200}, PlotPoints → 40,
 FrameLabel → {"Re √s  [MeV]", "Im √s  [MeV]"}, ImageSize → Large]
```

Out[222]=



Out[223]=



The pole must appear in the second Riemann sheets as $T_{\rm II}(s) \simeq \dfrac{g^2}{s - s_P} + \cdots$

Therefore, we must locate a zero in $\dfrac{1}{T_{\rm II}(s)} \simeq \frac{s - s_P}{g^2}$. Here it is easy to do that.

Definitions: $\sqrt{s_P} = m_\rho - i\,\Gamma_\rho$ or $s_P = m_\rho{}^2 - i\, m_\rho\, \Gamma_\rho$, with m_ρ and Γ_ρ the mass and the width of the ρ meson. (Here I am looking for the pole in the lower half of the complex plane just because it is customary, but recall that you can do it in either of the halves.)

In[224]:=
```
Wpole =
  W //. FindRoot[0 == 1/TII[W^2] //. FixedValuesFit //. fit["BestFitParameters"], {W, 775 - ⅈ 75}]
```

Out[224]= `762.33 - 68.632 ⅈ`

Now we can use our formula for the residues to make the numerical integration, which was:

$$g^2 = \frac{\rho}{2\pi} \int_0^{2\pi} T_{\rm II}\left(s_P + \rho\, e^{i\,\theta}\right) e^{i\,\theta}\, d\theta$$

```
In[253]:= ρsI = {2 m^2, m^2, 0.2 m^2} //. FixedValuesFit;(*I take different values for the radius ρ*)
gρSquared = ρsI/(2 π) NIntegrate[Exp[ⅈ θ] TII[s] //. FixedValuesFit //. fit["BestFitParameters"] //.
      {s → Wpole^2 + ρsI Exp[ⅈ θ]}, {θ, 0, 2 π}];
Print["Coupling (GeV^2) for different radius =", Abs[gρSquared] / 10^6]
Print["Coupling (dimensionless) for different radius =",
  Abs[Sqrt[3/(4 p2[Wpole^2]) gρSquared //. FixedValuesFit]]]

Coupling (GeV^2) for different radius ={5.72845, 5.72845, 5.72845}

Coupling (dimensionless) for different radius ={5.80487, 5.80487, 5.80487}
```

Omnes function $\Omega_1(s)$ and pion vector form factor $F_\pi^V(s)$

The Omnès function needs to be computed numerically. Since it is an integral, it is complicated to obtain. Here we use an interpolation from a numerically well computed function (with FORTRAN).

```
In[228]:= ImportedOmnesW =
   Import["/home/albala/Dropbox/Projects/2023/SalamancaSchool/FormFactor_Data/
      OmnesToInterpolate.dat"];
DataFormFactor = Import[
    "/home/albala/Dropbox/Projects/2023/SalamancaSchool/FormFactor_Data/CMD2-Data.dat"];
DataBelle = Import[
    "/home/albala/Dropbox/Projects/2023/SalamancaSchool/FormFactor_Data/BelleData.dat"];
(*DataFormFactorToFit=
   Join[DataFormFactor[[1;;20]],DataFormFactor[[29;;Length[DataFormFactor]]]];*)
DataFormFactorToFit = DataFormFactor[[All]];
DataFFPlot = ListLogPlot[
    {Table[{DataFormFactor[[i, 1]], Around[DataFormFactor[[i, 2]], DataFormFactor[[i, 3]]]},
      {i, 1, Length[DataFormFactor]}],
     Table[{1000 Sqrt[DataBelle[[i, 1]]], Around[DataBelle[[i, 2]],
         Sqrt[DataBelle[[i, 3]]^2 + DataBelle[[i, 4]]^2]]}, {i, 1, Length[DataBelle]}]},
    PlotStyle → {{Orange, PointSize[0.007]}, {Green, PointSize[0.007]}},
    PlotLegends → Placed[{"CMD-2", "Belle"}, {0.85, 0.85}],
    PlotRange → {{250, 1100}, {1, 50}}, AxesLabel → {" √s (GeV)", "|F_π^V(s)|^2"}];
Ω = Interpolation[
    Table[{ImportedOmnesW[[i, 1]]^2, ImportedOmnesW[[i, 2]] + ⅈ ImportedOmnesW[[i, 3]]},
     {i, 1, Length[ImportedOmnesW]}], InterpolationOrder → 1];
Ωp0 = Ω'[0];
```

```
FπV[s_] := (1 + (r2V/6 - Ωp0) s) Ω[s];
Expand[6 FπV'[0]]
fitFF = NonlinearModelFit[DataFormFactorToFit[[All, {1, 2}]],
   Abs[FπV[W^2] //. {r2V → r2VGeV / 10^6}]^2, {{r2VGeV, 23}}, W,
   Weights → 1 / DataFormFactorToFit[[All, 3]]^2, VarianceEstimatorFunction → (1 &)];
Print["Maximum Energy fitted (MeV)=",
 DataFormFactorToFit[[Length[DataFormFactorToFit], 1]]]
fitFF["ParameterTable"]
Print["χ^2/dof=", NumberForm[
  fitFF["ANOVATableSumsOfSquares"][[2]] / (Length[DataFormFactorToFit] - 1), {2, 1}]]
TableForm[{{"", "⟨r^2⟩_π^V (fm^2)", "", ""},
  {"Fit", NumberForm[(197.327/1000)^2 r2VGeV //. fitFF["BestFitParameters"], {4, 3}],
   "±", NumberForm[-(197.327/1000)^2 ((r2VGeV //. fitFF["BestFitParameters"]) -
        fitFF["ParameterConfidenceIntervals"][[1, 2]]), {4, 3}]},
  {"From Omnès", NumberForm[(197.327)^2 6 Re[Ωp0], {4, 3}], "", ""},
  {"PDG", NumberForm[(197.327/1000)^2 11.61, {4, 3}], "±", NumberForm[(197.327/1000)^2 0.28, {4, 3}]},
  {"Lattice", NumberForm[(197.327/1000)^2 10.50, {4, 3}], "±",
   NumberForm[(197.327/1000)^2 1.12, {4, 3}]}, {"Pich et al.",
   NumberForm[(197.327/1000)^2 11.04, {4, 3}], "±", NumberForm[(197.327/1000)^2 0.30, {4, 3}]}}]
r2Central = r2V //. {r2V → r2VGeV / 10^6} //. fitFF["BestFitParameters"];
r2Min =
  r2V //. {r2V → r2VGeV / 10^6} //. {r2VGeV → fitFF["ParameterConfidenceIntervals"][[1, 1]]};
r2Max =
  r2V //. {r2V → r2VGeV / 10^6} //. {r2VGeV → fitFF["ParameterConfidenceIntervals"][[1, 2]]};
Show[DataFFPlot, LogPlot[{Abs[FπV[W^2] //. {r2V → r2Central}]^2, Abs[FπV[W^2] //. {r2V → r2Min}]^2,
   Abs[FπV[W^2] //. {r2V → r2Max}]^2, Abs[FπV[W^2] //. {r2V → 6 Ωp0}]^2}, {W, 0, 1200},
  PlotStyle → {{Opacity[0.8], Black, Thick}, {Opacity[0.8], Gray, Dashed},
    {Opacity[0.8], Gray, Dashed}, {Opacity[0.8], Blue, Dashed}}, PlotLegends →
   Placed[{"Fit F_π^V(s)=p(s)Ω(s)", "1σ Bands", None, "F_π^V(s)=Ω(s)"}, {0.85, 0.85}],
  AxesLabel → {" √s (GeV)", "|F_π^V(s)|^2"}, ImageSize → Full]
Show[ListLogPlot[
```

```
{Table[{DataFormFactor[[i, 1]], Around[DataFormFactor[[i, 2]], DataFormFactor[[i, 3]]]},
   {i, 1, Length[DataFormFactor]}], Table[{1000 Sqrt[DataBelle[[i, 1]]], Around[DataBelle[[
      i, 2]], Sqrt[DataBelle[[i, 3]]^2 + DataBelle[[i, 4]]^2]]}, {i, 1, Length[DataBelle]}]},
  PlotStyle → {{Orange, PointSize[0.01]}, {Green, PointSize[0.01]}},
  PlotLegends → Placed[{"CMD-2", "Belle"}, {0.85, 0.85}],
  AxesLabel → {" √s (GeV)", "|F_π^V(s)|^2"}, PlotRange → {{0, 1800}, {0.01, 50}}],
 LogPlot[{Abs[FπV[W^2] //. {r2V → r2Central}]^2, Abs[FπV[W^2] //. {r2V → 6 Ωp0}]^2},
  {W, 0, 1800}, PlotStyle → {{Black, Thick}, {Blue, Dashed}},
  PlotLegends → Placed[{"Fit F_π^V(s)=p(s)Ω(s)", "F_π^V(s)=Ω(s)"}, {0.85, 0.85}],
  AxesLabel → {" √s (GeV)", "|F_π^V(s)|^2"},
  PlotRange → {{0, 1800}, {0.01, 50}}], ImageSize → Full]
```

Out[236]= (0. + 0. i) + (1. + 0. i) r2V

Maximum Energy fitted (MeV)=961.52

Out[239]=

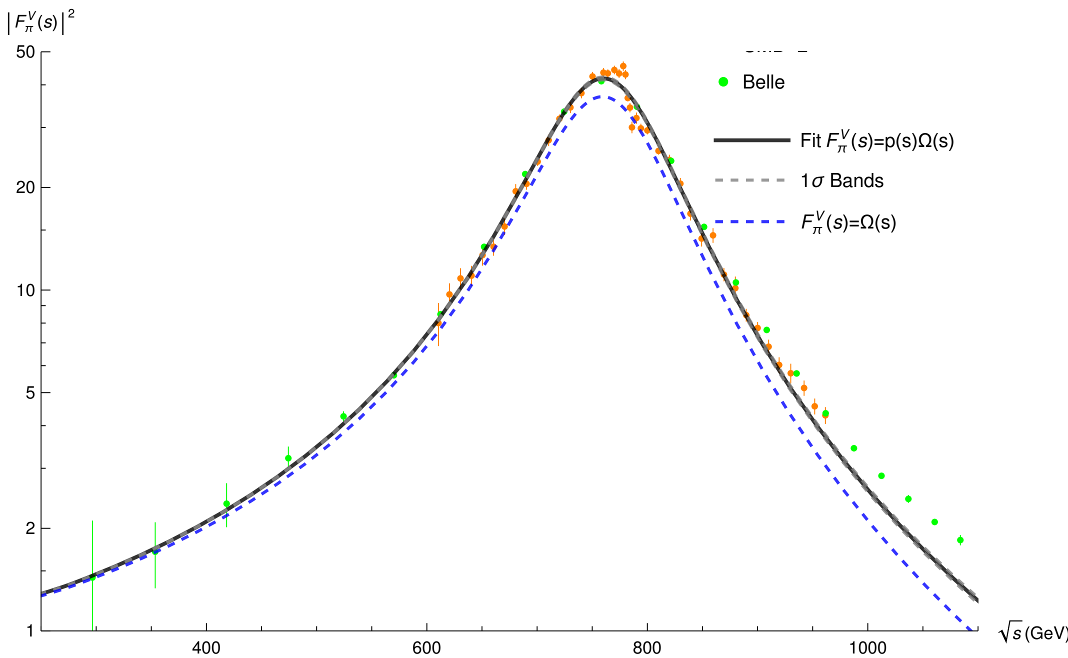
	Estimate	Standard Error	t-Statistic	P-Value
r2VGeV	11.0202	0.0268534	410.383	2.62892×10^{-77}

χ^2/dof=4.5

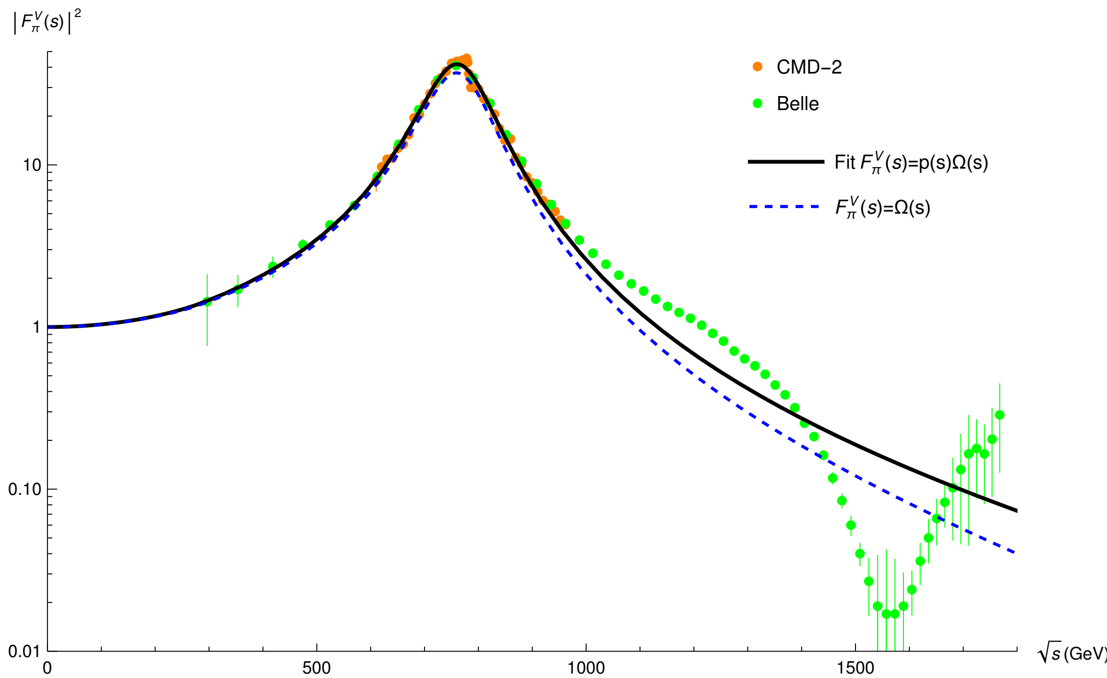
Out[241]//TableForm=

	$\langle r^2\rangle_\pi^V$ (fm^2)		
Fit	0.429	±	0.002
From Omnès	0.403		
PDG	0.452	±	0.011
Lattice	0.409	±	0.044
Pich et al.	0.430	±	0.012

Out[245]=

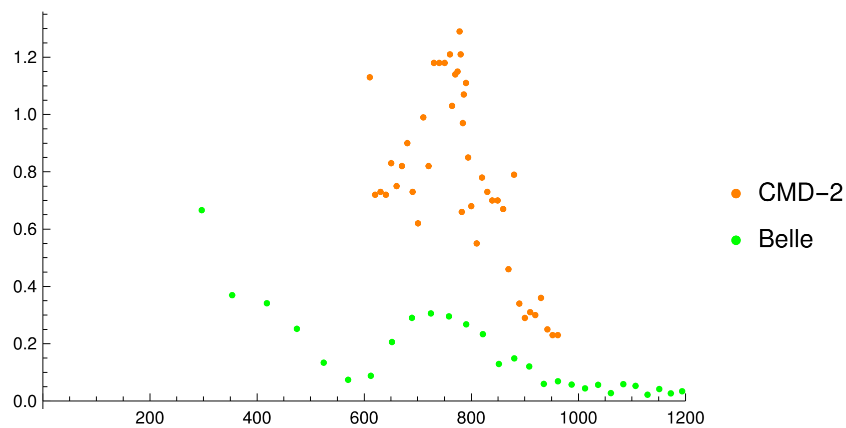


Out[246]=



In[247]:=
```
ListPlot[
 {Table[{DataFormFactor[[i, 1]], DataFormFactor[[i, 3]]}, {i, 1, Length[DataFormFactor]}],
  Table[{1000 Sqrt[DataBelle[[i, 1]]], Sqrt[DataBelle[[i, 3]]^2 + DataBelle[[i, 4]]^2]},
   {i, 1, Length[DataBelle]}]},
 PlotStyle → {{Orange, PointSize[0.01]}, {Green, PointSize[0.01]}},
 PlotLegends → {"CMD-2", "Belle"}, PlotRange → {{0, 1200}, Automatic}]
ListPlot[
 {Table[{DataFormFactor[[i, 1]], Around[DataFormFactor[[i, 2]], DataFormFactor[[i, 3]]]},
   {i, 1, Length[DataFormFactor]}],
  Table[{1000 Sqrt[DataBelle[[i, 1]]], Around[DataBelle[[i, 2]],
      Sqrt[DataBelle[[i, 3]]^2 + DataBelle[[i, 4]]^2]]}, {i, 1, Length[DataBelle]}]},
 PlotStyle → {{Orange, PointSize[0.01]}, {Green, PointSize[0.01]}},
 PlotLegends → {"CMD-2", "Belle"},
 PlotRange → {{600, 900}, Automatic}]
```

Out[247]=



Out[248]=