

Example-03: phase space

```
1 begin
2     using Pkg
3     Pkg.activate(mktempdir())
4     Pkg.add([
5         Pkg.PackageSpec(url="https://github.com/JuliaHEP/FourVectors.jl.git"),
6         Pkg.PackageSpec(url="https://github.com/mmikhasenko/RamboOnDiet.jl.git"),
7         Pkg.PackageSpec("HadronicLineshapes"),
8         Pkg.PackageSpec("DataFrames"),
9         Pkg.PackageSpec("Plots"),
10    ])
11    #
12    using FourVectors
13    using RemboOnDiet
14    using HadronicLineshapes
15    using Plots
16    using DataFrames
17 end
```

```
1 theme(:boxed)
```

amplitude (generic function with 1 method)

```
1 function amplitude(x::Array{<:FourVector}; pars)
2     p1, p2, p3 = x
3      $\sigma_1$  = mass2(p2+p3)
4      $\sigma_2$  = mass2(p3+p1)
5      $\sigma_3$  = mass2(p1+p2)
6     #
7     amplitude_model( $\sigma_1, \sigma_2, \sigma_3$ ; pars)
8 end
```

default_pars =

(m1 = 0.8, Γ_1 = 0.15, c1 = 1.91067+0.59104im, m2 = 1.2, Γ_2 = 0.05, c2 = 1.0, m3 = 1.6, Γ_3 = 0.

```
1 default_pars = (
2     m1 = 0.8,  $\Gamma_1$  = 0.15, c1 = 2.00*cis(0.3),
3     m2 = 1.2,  $\Gamma_2$  = 0.05, c2 = 1.0,
4     m3 = 1.6,  $\Gamma_3$  = 0.05, c3 = 3.0)
```

amplitude_model (generic function with 1 method)

```
1 amplitude_model( $\sigma_1::Real, \sigma_2::Real, \sigma_3::Real$ ; pars) = let
2     (; m1,  $\Gamma_1, c1$ ) = pars
3     (; m2,  $\Gamma_2, c2$ ) = pars
4     (; m3,  $\Gamma_3, c3$ ) = pars
5     #
6     c1 * BreitWigner(m1, $\Gamma_1$ )( $\sigma_1$ ) +
7     c2 * BreitWigner(m2, $\Gamma_2$ )( $\sigma_2$ ) +
8     c3 * BreitWigner(m3, $\Gamma_3$ )( $\sigma_3$ )
9 end
```

unnormalized_density (generic function with 1 method)

```
#  $x \in \mathbb{R}^{12}$ ,  $p \in \mathbb{R}^6 \times \mathbb{C}^3$   
unnormalized_density(x; pars) = abs2(amplitude(x; pars))
```

What is \vec{x}

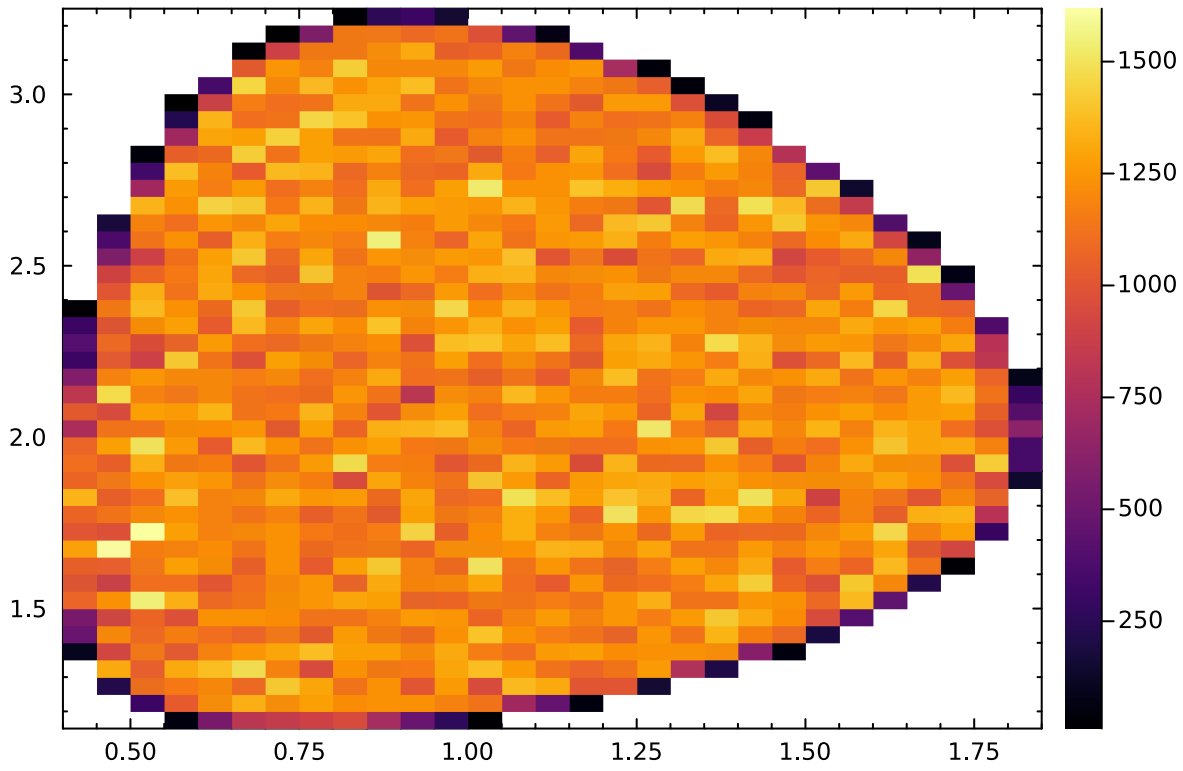
unnormalized_density is defined on a x which is 3x4 dimensions

```
p1 = [p1x,p1y,p1z,E1],  
p2 = [p2x,p2y,p2z,E2],  
p3 = [p3x,p3y,p3z,E3]
```

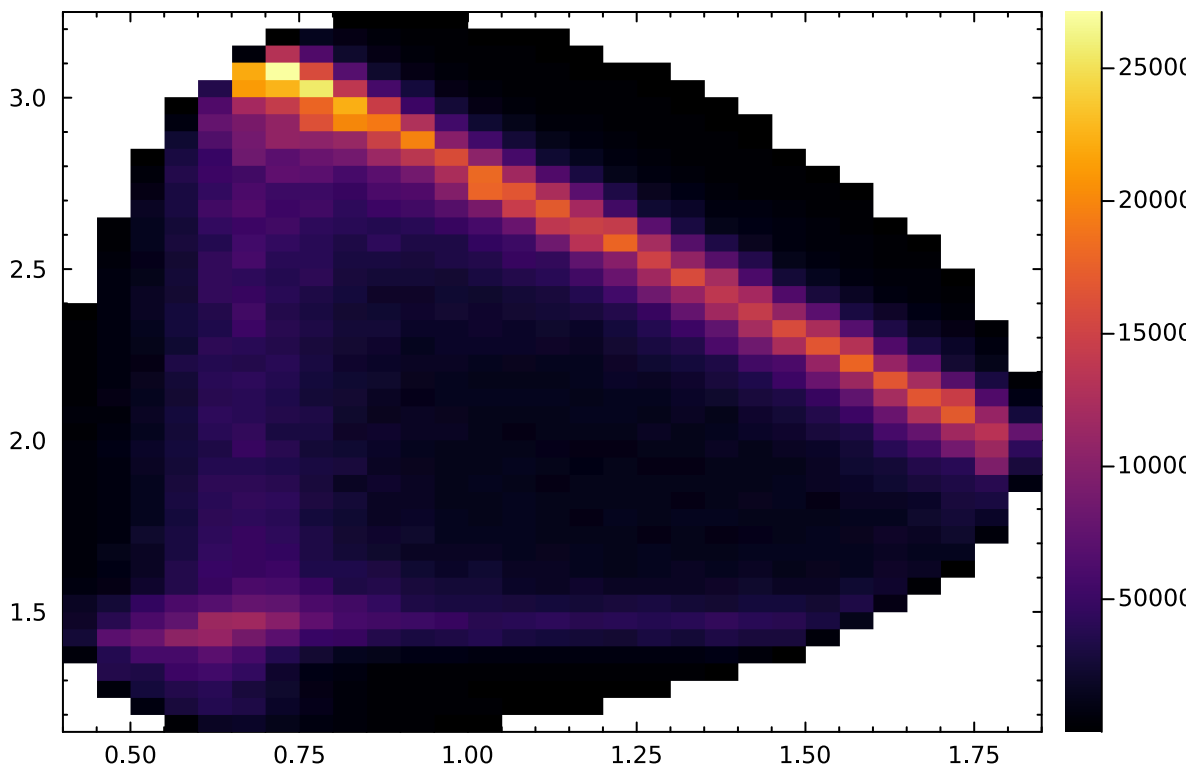
Evaluating the density on MC sample

```
data = let N = 100_000  
  masses = [0.93827208816, 0.493677, 0.13957039]  
  generator = PhaseSpaceGenerator(masses, 2.28646)  
  df = [rand(generator) for _ in 1:N] |> DataFrame  
  #  
  select(df, :momenta => ByRow() do (p1, p2, p3)  
     $\sigma_1, \sigma_2 = \text{mass2}(p2+p3), \text{mass2}(p3+p1)$   
    (; p1, p2, p3,  $\sigma_1, \sigma_2$ )  
    end => AsTable, :weight)  
end;
```

```
density = select(data, [:p1,:p2,:p3] => ByRow() do p1, p2, p3  
  unnormalized_density([p1,p2,p3]; pars=default_pars)  
end => :weight);
```



```
histogram2d(data.σ1, data.σ2, bins=50; weights=data.weight)
```



```
histogram2d(data.σ1, data.σ2, bins=50;  
weights=data.weight .* density.weight)
```