

## Example 2: 2D Extended Model

---

```
1 begin
2   using Pkg
3   Pkg.activate(mktempdir())
4   Pkg.add([
5     Pkg.PackageSpec("NumericalDistributions"),
6     Pkg.PackageSpec("HadronicLineshapes"),
7     Pkg.PackageSpec(; url="https://github.com/JuliaHEP/DistributionsHEP.jl.git",
8       rev="codex/extended-mixture-model"),
9     Pkg.PackageSpec("Distributions"),
10    Pkg.PackageSpec("Plots"),
11  ])
12  #
13  using NumericalDistributions
14  using HadronicLineshapes
15  using DistributionsHEP
16  using Distributions
17  using Plots
18 end
```

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

```
1 begin
2   # this code should go to NumericalDistributions
3   #
4   struct Abs2 <: AbstractFlexFunc
5     F
6   end
7   (o::Abs2)(x) = abs2(o.F(x))
8
9   struct ofAbs2 <: AbstractFlexFunc
10    F
11  end
12  (o::ofAbs2)(x) = o.F(x^2)
13 end
```

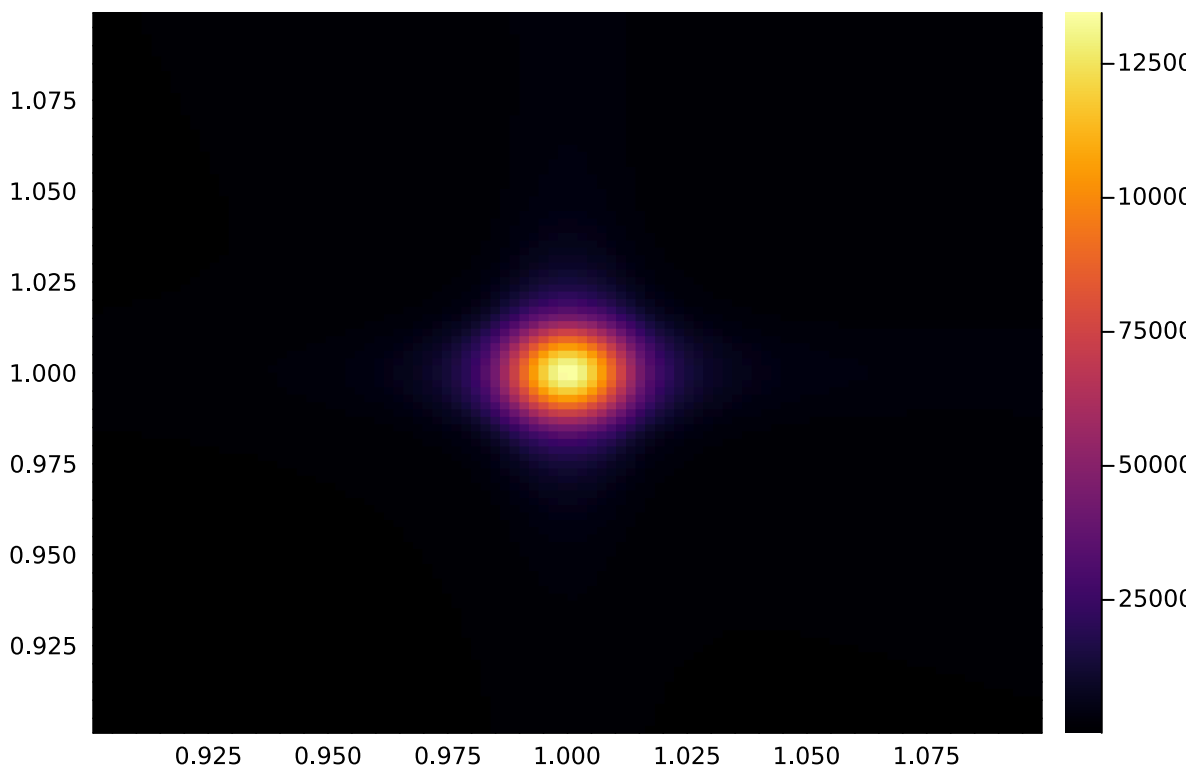
(0.9, 1.1)

```
1 lb, ub = 0.9, 1.1
```

```

model =
  ExtendedMixtureModel([Distributions.Product{Distributions.Continuous, DistributionsHEP.DoubleCrystalBallTail{Float64}}(12.9518, 4.0, 150.0, 0.985)
1  model = let
2     y_phiphi = 10_000
3     y_mixed = 1_500
4     y_kkkk = 3_000
5
6     signal = DoubleCrystalBall(1.0, 0.01, 1.5, 4.0, 1.5, 4.0)
7     background = Chebyshev([1.0,0.5], lb, ub)
8
9     phiphi = product_distribution([signal, signal])
10    mixed = MixtureModel(
11      [
12        product_distribution([signal, background]),
13        product_distribution([background, signal]),
14      ],
15      [0.5, 0.5],
16    )
17    kkkk = product_distribution([background, background])
18
19    ExtendedMixtureModel([phiphi, mixed, kkkk], [y_phiphi, y_mixed, y_kkkk])
20  end

```



```

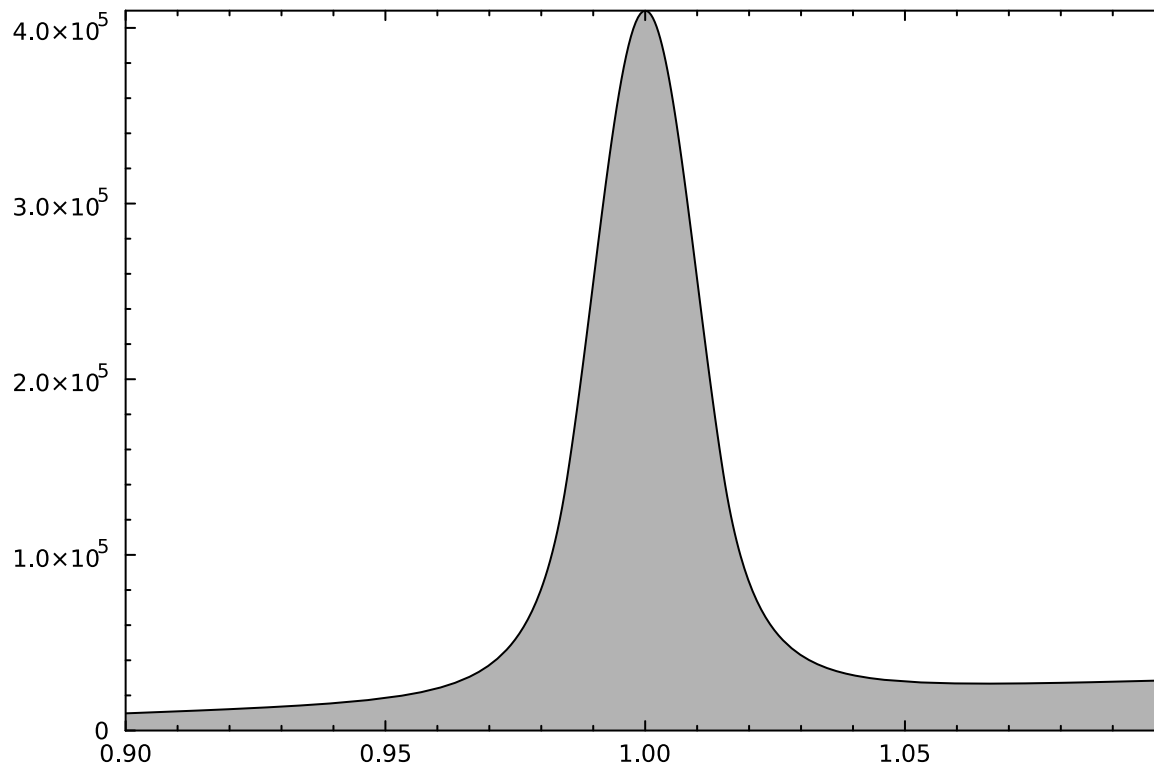
let
  heatmap(
    range(lb, ub, 100)[2:end-1],
    range(lb, ub, 100)[2:end-1],
    (x,y)->model([x,y])
  )
end

```

```

begin
  proj1 = DistributionsHEP.marginalize(model, 1)
  proj2 = DistributionsHEP.marginalize(model, 2)
end;

```



```
let  
  plot(x->proj1(x), lb, ub, fill=0, fillalpha=0.3)  
end
```