Kimenatics of D^x D π

In this notebook, we take a single kinematic point,

```
Dst: 2.0085299,0.0570074,-0.026685,0.0479813
D: 1.8967276,-0.108349,-0.056907,-0.295222
Pi: 0.3153471, 0.1034199, 0.0873037,0.2482869
```

and compute all helicity angles

```
begin
    using Pkg
    Pkg.activate(mktempdir())
    Pkg.add([
        Pkg.PackageSpec(url="https://github.com/JuliaHEP/LorentzVectorBase.jl"),
        Pkg.PackageSpec(url="https://github.com/mmikhasenko/FourVectors.jl"),
        Pkg.PackageSpec(url="https://github.com/mmikhasenko/DecayAngles.jl"),
        Pkg.PackageSpec("Parameters"),
        Pkg.PackageSpec("DataFrames")])
    #
    using DecayAngles
    using FourVectors
    using DataFrames
    using Parameters
end
```

```
pure_B (generic function with 1 method)
```

```
function pure_B(p::FourVector, p_ref::FourVector)
    @unpack cosθ, ϕ = spherical_coordinates(p_ref)
    θ = acos(cosθ)
    γ = boost_gamma(p_ref)
    p |> Rz(-ϕ) |> Ry(-θ) |> Bz(-γ) |> Ry(θ) |> Rz(ϕ)
end
```

```
pure_B (generic function with 2 methods)
```

```
function pure_B(system::NamedTuple)
    ptot = collect(system) |> sum
    map(system) do p
        pure_B(p, ptot)
    end |> NamedTuple{keys(system)}
end
```

```
helicity_angles (generic function with 1 method)
```

```
function helicity_angles(four_vectors_rf, topology)
    momenta_dict = Dict(pairs(four_vectors_rf))
    tree_empty = DecayNode(topology);
    tree_with_particle_order = add_indices_order(tree_empty);
    tree_with_four_vectors = add_transform_through(
        HelicityTransformation, tree_with_particle_order, momenta_dict);
    #
    decay_angles(tree_with_four_vectors)
end
```

```
four_vectors_nt =
  (Dst = FourVectors.FourVector{Float64}: [0.0570074, -0.026685, 0.0479813, 2.00853], D =
```

```
four_vectors_nt = (
    Dst = FourVector(0.0570074,-0.026685,0.0479813; E=2.0085299),
    D = FourVector(-0.108349,-0.056907,-0.295222; E=1.8967276),
    Pi = FourVector(0.1034199, 0.0873037,0.2482869; E=0.3153471)
)
```

```
[2.00697, 1.86961, 0.13957]
```

```
## test if masses are reasonable
four_vectors_nt |> collect .|> mass
```

```
4.2202815272158025
```

```
# sqrt(s)
collect(four_vectors_nt) |> sum |> mass
```

```
four_vectors_rf =
  (Dst = FourVectors.FourVector{Float64}: [0.0322264, -0.0284512, 0.0474835, 2.00799], D =
```

```
# pure boost to the rest frame
four_vectors_rf = pure_B(four_vectors_nt)
```

	names	m	child1_θ	child1_φ
1	((:Pi, :D), :Dst)	4.22028	2.40584	2.41833
2	(:Pi, :D)	2.21136	2.36202	-2.43617
3	((:D, :Dst), :Pi)	4.22028	2.65214	-2.42315
4	(:D, :Dst)	3.89619	0.23836	-1.18248
5	((:Dst, :Pi), :D)	4.22028	0.453726	0.418314
6	(:Dst, :Pi)	2.29853	2.97652	-1.69129

```
begin
    vcat(
        helicity_angles(four_vectors_rf, ((:Pi, :D), :Dst)),
        helicity_angles(four_vectors_rf, ((:D, :Dst), :Pi)),
        helicity_angles(four_vectors_rf, ((:Dst, :Pi), :D))
    ) |> DataFrame
end
```