

```
• using Graphs , GraphMakie , NetworkLayout , CairoMakie
```

```
gr = {5, 12} directed simple Int64 graph
```

```
• gr = smallgraph(:house) |> DiGraph
```

## transform\_tangents

Rotates tangents of Bezier graph about the edge angle

# Arguments

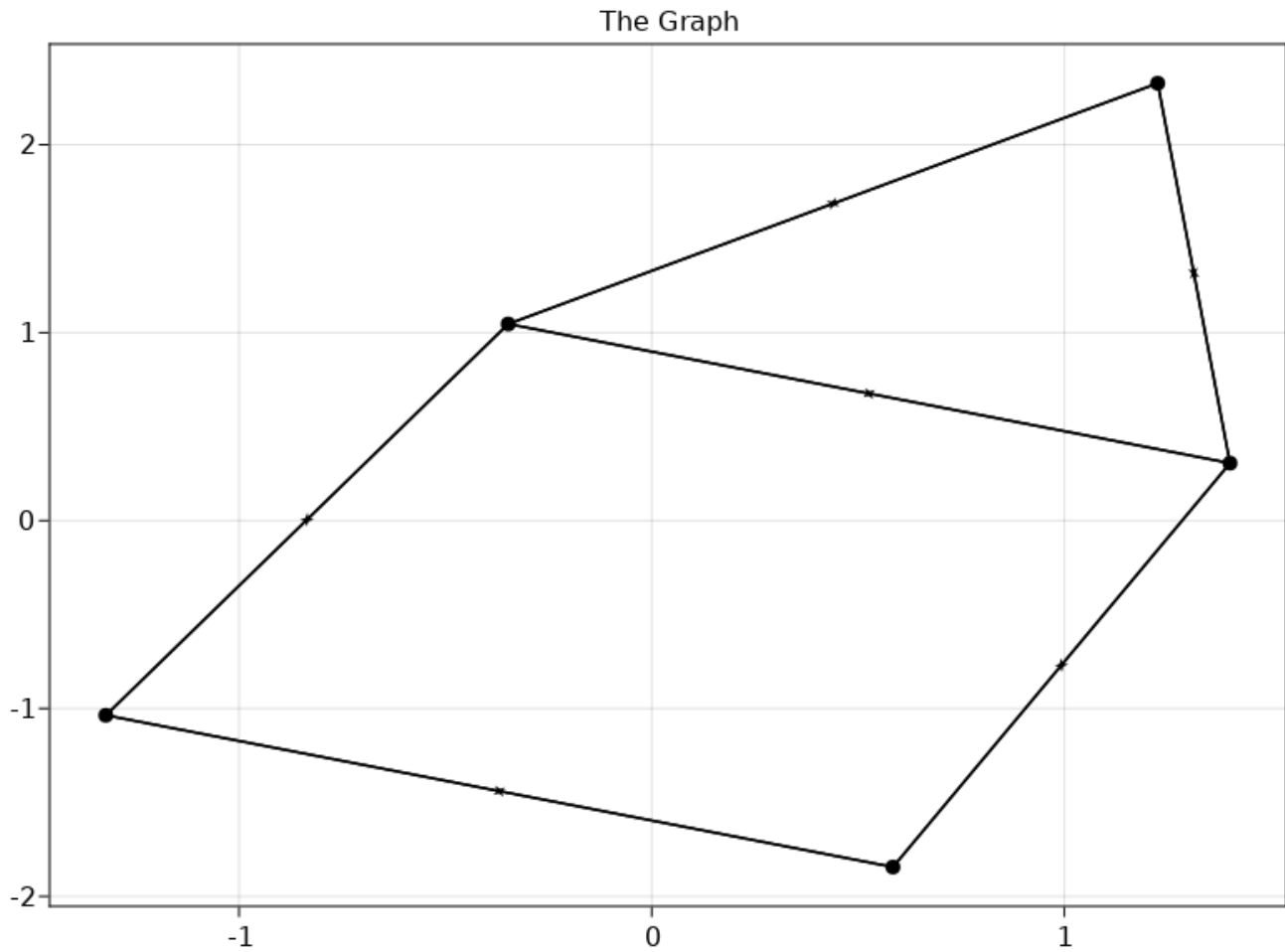
- Graph
- Tangents
- function fun(x) to return node position upon call

```
• """
• Rotates tangents of Bezier graph about the edge angle
• # Arguments
• - Graph
• - Tangents
• - function fun(x) to return node position upon call
• """
• function transform_tangents(gr::AbstractGraph, ts, nposfun::Function)
•     ts2 = Dict{Int,Any}{}
•     for (i,e) in enumerate(edges(gr))
•         θ = angletox(nposfun(e.dst) - nposfun(e.src))
•         ts2[i] = Tuple(Tuple(rotate2D(θ) * [tel for tel in t]) for t in ts[i])
•     end
•     return ts2
• end
```

## angletox

Returns angle of vector to x axis

```
• """
• Returns angle of vector to x axis
• """
• function angletox(a)
•     θ = atan(a[2]/a[1])
•     a[1] < 0 ? pi + θ : θ
```

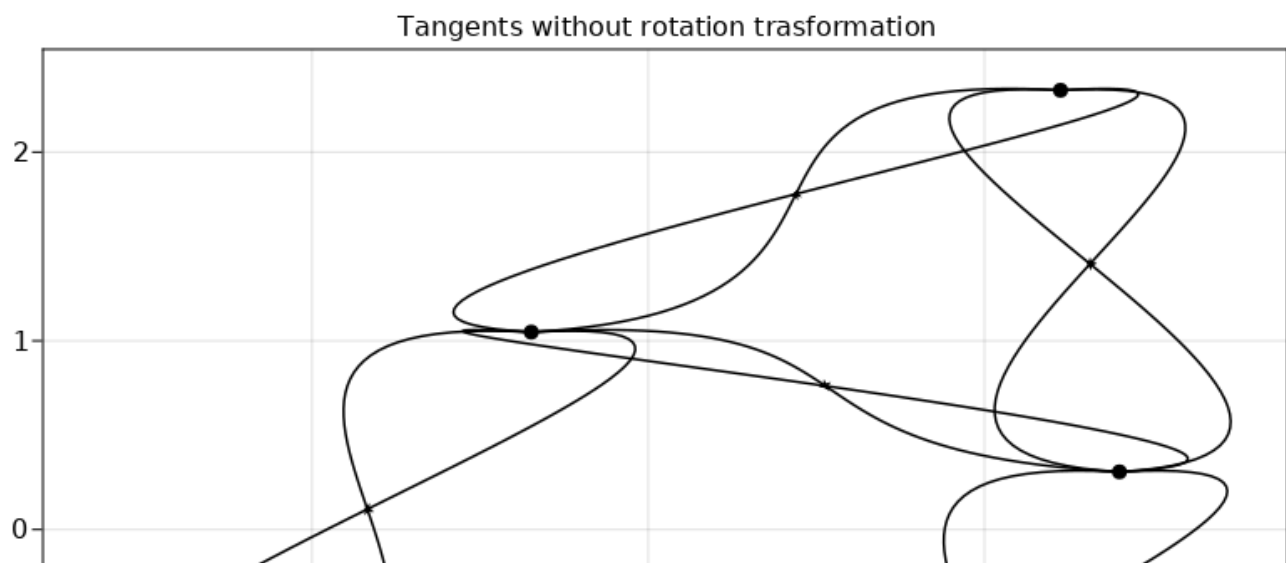


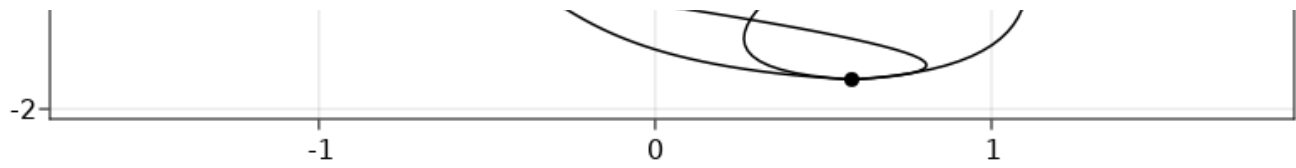
```
• f, a, p = graphplot(gr) ; a.title = "The Graph"; f
```

```
tangs =
```

```
Dict(5 ⇒ ((10, 1), (10, -1)), 12 ⇒ ((10, 1), (10, -1)), 8 ⇒ ((10, 1), (10, -1)), 1 ⇒
```

```
• tangs = Dict(i => ((10,1),(10,-1)) for (i,_) in enumerate(edges(gr)))
```

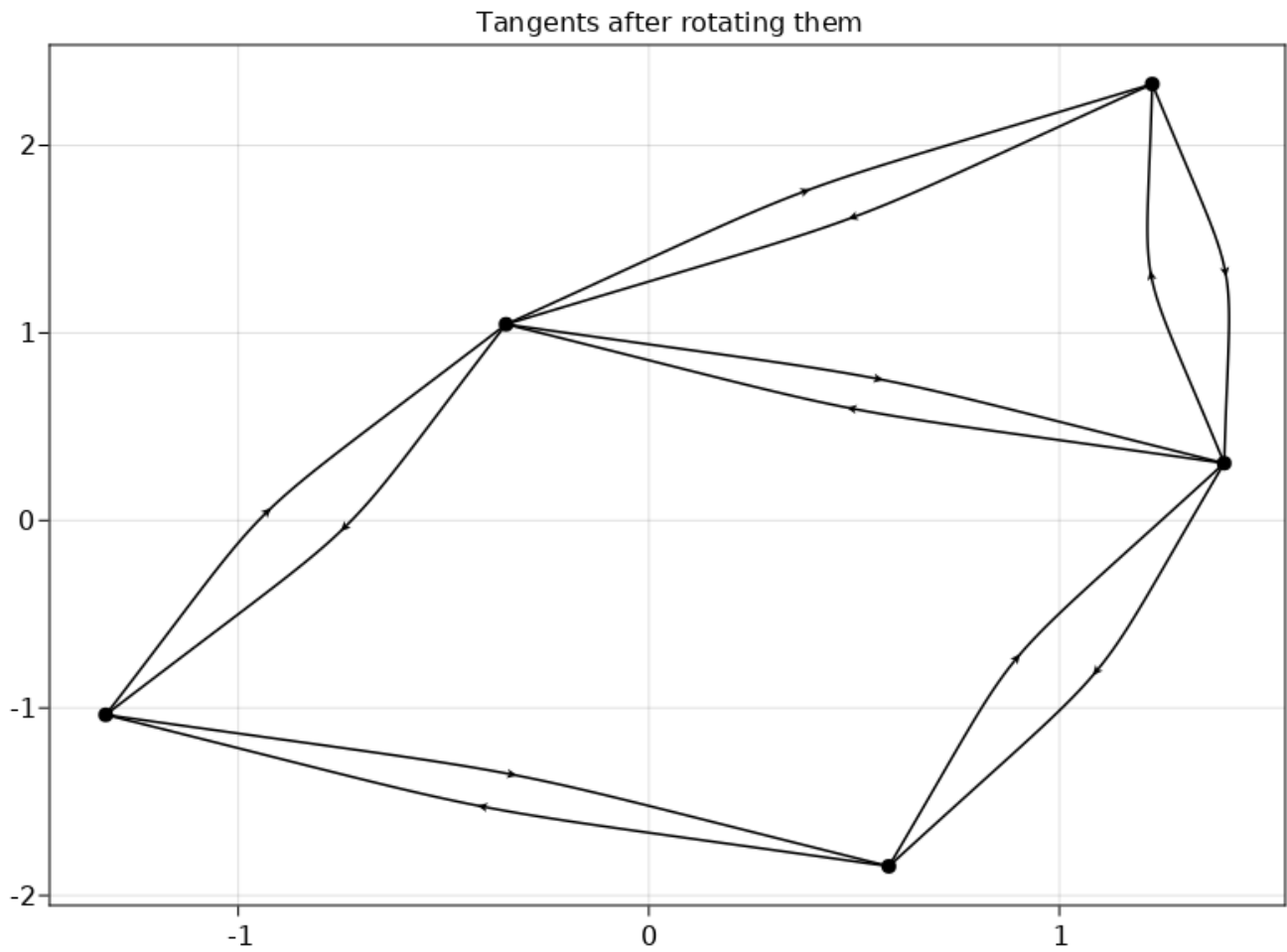




```

• begin
•   local ft, at, pt = graphplot(gr, tangents = tangs)
•   at.title = "Tangents without rotation trasformation"
•   ft
• end

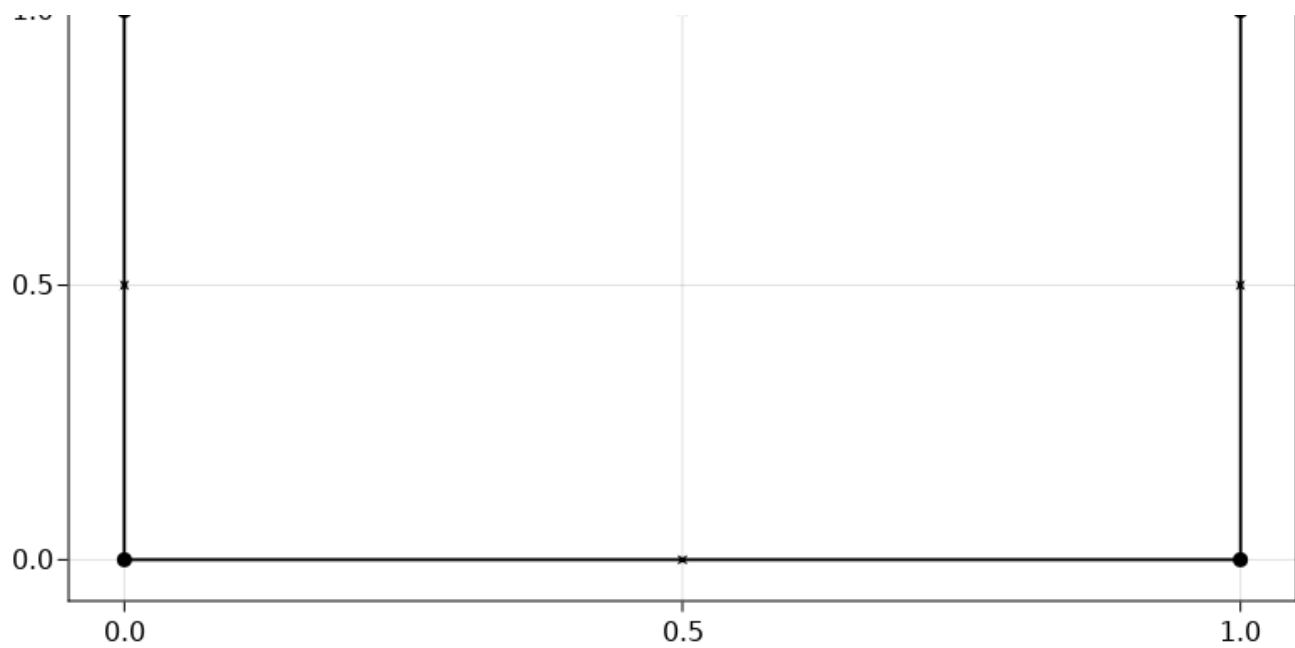
```



```

• begin

```



```
• f2, a2, p2 = graphplot(gr; layout = fixed_layout)
```

nposfun2 (generic function with 1 method)

```
• nposfun2(v::Int) = p2[:node_pos].val[v]
```

