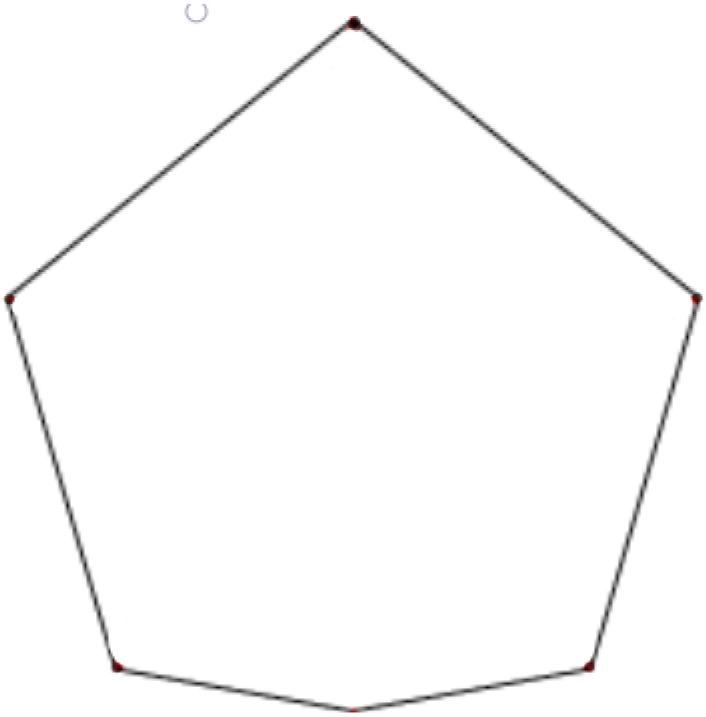
In [49]:

```
# rectangular coordinates

using JuMP, Ipopt
n = 6
m = Model(solver = IpoptSolver(print_level=0))

@variable(m, x[1:n] )
@variable(m, y[1:n] )
#@variable(m, dist[1])

@NLobjective(m, Max, 0.5*sum( x[i]*y[i]-y[i]*x[i+1] for i=1
:n-1)  +  0.5*(x[n]*y[1]-y[n]*x[1]) )

for i = 1:n
    for j = 1:n
        if i != j && x[i] != x[j]   && y[i] != y[j]
            @NLconstraint(m, sqrt((x[j] - x[i])^2 + (y[j] -
y[i])^2 ) <= 1);
        end
    end
end

# add ordering constraint to the vertices
for i = 1:n-1
    @constraint(m, x[i]*y[i+1]-y[i]*x[i+1] >= 0 )
end
@NLconstraint(m, x[n]*y[1]-y[n]*x[1] >= 0 )


srand(0)
setvalue(x,rand(n))
setvalue(y,rand(n))

#setvalue(x[1],0)
#setvalue(y[1],0)

status = solve(m)
println(status)
println("Optimal area: ", getobjectivevalue(m))
getvalue([x y])

using PyPlot
```

```
xopt = getvalue([x; x[1]])
yopt = getvalue([y; y[1]])
t = linspace(0,2π,100)
figure(figsize=[5,5])
#plot( cos.(t), sin.(t), "b-" )
plot( xopt, yopt, "r.-" )
axis("equal");
axis("off");
```



```
Optimal
Optimal area: 0.649519
```