

```

require 'gnuplot'
Gnuplot.open do |gp|
  Gnuplot::Plot.new(gp) do |plot|
    plot.terminal 'png size 1200,800'
    (800..805).each do |g|
      p g
    end
    # plot.style "line 1 lc rgb 'gold'"

    plot.noborder
    plot.nokey
    plot.polar
    plot.noraxis
    plot.nortics
    plot.noxtics
    plot.noytics
    plot.size '0.66, 1.0'
    plot.samples '1600'
    g = 13
    plot.output File.expand_path("../r#{g}.png", __FILE__)
    x = [], y = []
    # (Math::PI/24..Math::PI*24).step(Math::PI/24) do |i|
    #   p i
    #   r = 36 * Math.sin(g * i)
    #   x << r * Math.cos(i)
    #   y << r * Math.sin(i)
    # end
    # plot.data = [
    # Gnuplot::DataSet.new( [y,x] ) { |ds|
    #   ds.with = "point"
    #   ds.with = "circle"
    #   ds.with = "lines"
    #   ds.linewidth = 2
    #   ds.with = "linespoints"
    #   ds.title = "step Pi/12"
    # }
    # ]

    (0..Math::PI * 144_000 / 12).step((3 - Math.sqrt(5))) do |i|
      r = 1 * Math.sin(g * i)
      x << r * Math.cos(i)
      y << r * Math.sin(i)
    end
    plot.data = [

```

```

    Gnuplot::DataSet.new([y, x]) do |ds|
      # ds.with = "dot"
      ds.with = 'point'
      # ds.with = "circle"
      # ds.with = "lines"
      ds.linewidth = 1
      # ds.with = "linespoints"
      ds.title = 'step Pi/12'
    end
  ]
end
# end

end
# (0..Math::PI * 2).step(0.000175) do |i|
#   r = 352 * Math.sin(g * i)
#   x = (r * Math.cos(i))# + @centerx
#   y = (r * Math.sin(i))# * ar + @centery
# end
# def point(x,y,color, size)
#   stroke color
#   (0..size).each do |i|
#     line x,y,x+size,y+i
#   end
# end
# end

# stroke bg_color
# line x,y+1,x+1,y+1
# height, width = 760.0, 1024.0
# Shoes.app :height => 768, :width => 1024, :title => "      Roseland Revisited" do
#   background white
#   @centery, @centerx = height / 2 - 30, width / 2
#   @radius = @centerx / @centery * @centerx

#   ar = width / height

#   g = 965

#   (0..Math::PI * 2).step(0.000175) do |i|
#     r = 352 * Math.sin(g * i)
#     x = (r * Math.cos(i)) + @centerx
#     y = (r * Math.sin(i)) * ar + @centery
#     point x.to_i.abs, y.to_i.abs, tan, 3
#     puts "y = #{y.to_i.abs}, x = #{x.to_i.abs}"
#   end
# end

```

```
# r > 256 ? c = (r.to_i.abs + 1) : c = 256
# c = c * 1024
# point x.to_i, y.to_i, lawngreen, 3#"#a" + c.to_s(16)
# puts "#{x.to_i}, #{y.to_i}"
# end
```