

```

using Gadfly
using Colors

##### Function for saving plots #####
# Input:
# fileName = Name of the file where the plot is to be stored
#             (with or without extension)
# f_e = Array containing function evaluations as columns for each solver
# err = Array containing errors as columns for each solver
# solverNames = Array containing the names of solvers used in respective order
# plotSize = size of the plot to be created
#
# Values have been tuned for a graph similar to the one in
# Solving Ordinary Differential Equations I by
# Hairer, Ernst, Nørsett, Syvert P., Wanner, Gerhard
# page: 252
#####
function savePlotPNG(fileName,f_e,err,solverNames,
    plotSize=[30cm,30cm])

    numOfLayers = length(solverNames);

    if !contains(fileName, ".")
        fileName = string(fileName, ".png");
    end

    plotColorsHex = ["#4D4D4D", "#5DA5DA", "#FAA43A", "#60BD68",
        "#F17CB0", "#B2912F", "#B276B2", "#DECF3F", "#F15854"];
    plotColors = [parse(Colorant, c) for c in plotColorsHex];

    majorFontSize = 24pt;
    minorFontSize = 20pt;
    pointSize = 5pt;

    myplot = plot(Scale.x_log10, Scale.y_log10,
        Coord.cartesian(xflip=true),
        Guide.manual_color_key("Legend", solverNames, plotColorsHex[1:numOfLayers]),
        Guide.xlabel("error"), Guide.ylabel("#Function Evaluations"),
        Theme(major_label_font_size=majorFontSize, panel_stroke=colorant"black",
            minor_label_font_size=minorFontSize, key_title_font_size=majorFontSize,
            key_label_font_size=minorFontSize, key_position=:top, key_max_columns=1));

    for i = 1:numOfLayers
        push!(myplot, layer(x=err[:,i], y=f_e[:,i], Geom.point, Geom.path,
            Theme(default_color=plotColors[i], default_point_size=pointSize)));
    end

    draw(PNG(fileName, plotSize[1], plotSize[2]), myplot)
    return nothing
end

```