```julia
using PyPlot;

# plot a solution at three instances in time
function plot_solution(xs, ts, us::AbstractMatrix{Float64}; t::String="",
                       fname::String="", show_plot::Bool=false)

  plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.1f\$", ts[1]));
  plot(xs, us[:, div(length(ts), 2)], "k--"; label=@sprintf("\$t = %.1f\$",
                                                     ts[div(length(ts), 2)]));
  plot(xs, us[:, end], "k-."; label=@sprintf("\$t = %.1f\$", ts[end]));
  legend(; loc=3);
  if t != ""
    title(t);
  end
  xlabel("\$x\$");
  ylabel("\$u\$");
  if show_plot
    show();
  end
  if fname != ""
    savefig(fname);
  end
  clf();
end

# plot solution at three instances in time vs. an analytical solution
function plot_solution(xs, ts, us::AbstractMatrix{Float64}, asoln::Function;
                       t::String="", fname::String="", show_plot::Bool=false)

  plot(xs, us[:, 1], "kx"; label=@sprintf("\$t = %.1f\$", ts[1]));
  plot(xs, map(x -> asoln(x, 0), xs), "k-"; label=@sprintf("\$t = %.1f\$", ts[1]));
  plot(xs, us[:, div(length(ts), 2)], "ko"; label=@sprintf("\$t = %.1f\$",
                                                     ts[div(length(ts), 2)]));
  plot(xs, map(x -> asoln(x, ts[div(length(ts), 2)]), xs), "k--";
               label=@sprintf("\$t = %.1f\$", ts[div(length(ts), 2)]));
  plot(xs, us[:, end], "kv"; label=@sprintf("\$t = %.1f\$", ts[end]));
  plot(xs, map(x -> asoln(x, ts[end]), xs), "k-.";
       label=@sprintf("\$t = %.1f\$", ts[end]));
  legend(; loc=3);
  if t != ""
    title(t);
  end
  xlabel("\$x\$");
  ylabel("\$u\$");
  if show_plot
    show();
  end
  if fname != ""
    savefig(fname);
  end
  clf();
end
```