

```

1  using PyPlot;
2
3  # plot a solution at three instances in time
4  function plot_solution(xs, ts, us::AbstractMatrix{Float64}; t::String="",
5                        fname::String="", show_plot::Bool=false)
6
7      plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.1f\$", ts[1]));
8      plot(xs, us[:, div(length(ts), 2)], "k-"; label=@sprintf("\$t = %.1f\$",
9                        ts[div(length(ts), 2)]));
10     plot(xs, us[:, end], "k-"; label=@sprintf("\$t = %.1f\$", ts[end]));
11     legend(; loc=3);
12     if t != ""
13         title(t);
14     end
15     xlabel("\$x\$");
16     ylabel("\$u\$");
17     if show_plot
18         show();
19     end
20     if fname != ""
21         savefig(fname);
22     end
23     clf();
24 end
25
26 # plot solution at three instances in time vs. an analytical solution
27 function plot_solution(xs, ts, us::AbstractMatrix{Float64}, asoln::Function;
28                       t::String="", fname::String="", show_plot::Bool=false)
29
30     plot(xs, us[:, 1], "kx"; label=@sprintf("\$t = %.1f\$", ts[1]));
31     plot(xs, map(x -> asoln(x, 0), xs), "k-"; label=@sprintf("\$t = %.1f\$", ts[1]));
32     plot(xs, us[:, div(length(ts), 2)], "ko"; label=@sprintf("\$t = %.1f\$",
33                       ts[div(length(ts), 2)]));
34     plot(xs, map(x -> asoln(x, ts[div(length(ts), 2)]), xs), "k-";
35           label=@sprintf("\$t = %.1f\$", ts[div(length(ts), 2)]));
36     plot(xs, us[:, end], "kv"; label=@sprintf("\$t = %.1f\$", ts[end]));
37     plot(xs, map(x -> asoln(x, ts[end]), xs), "k-";
38           label=@sprintf("\$t = %.1f\$", ts[end]));
39     legend(; loc=3);
40     if t != ""
41         title(t);
42     end
43     xlabel("\$x\$");
44     ylabel("\$u\$");
45     if show_plot
46         show();
47     end
48     if fname != ""
49         savefig(fname);
50     end
51     clf();
52 end

```