

```

1  using PyPlot;
2
3  u0(x) = if abs(x) <= 0.5;
4      (cos(pi * x))^2;
5  else
6      0;
7  end;
8
9  const x0 = -1.0;
10 const x1 = 3.0;
11
12 const t0 = 0.0;
13 const t1 = 2.4;
14
15 const lambda = 0.8;
16
17 for h in [1./10.; 1./20.; 1./40.]
18
19     println("Forward-Time Backward-Space, h = $h");
20
21     xs = collect(x0:h:(x1+eps()));
22     k = h * lambda;
23     ts = collect(t0:k:(t1+eps()));
24     us = zeros(length(xs), length(ts));
25     us[:, 1] = map(u0, xs); # initial conditions
26
27     for n in 1:length(ts)-1, m in 2:length(xs)
28         us[m, n+1] = us[m, n] - lambda * (us[m, n] - us[m-1, n]);
29     end
30
31     plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.3f\$", ts[1]));
32     plot(xs, us[:, div(length(ts), 2)], "k--"; label=@sprintf("\$t = %.3f\$",
33         ts[div(length(ts), 2)]));
34     plot(xs, us[:, end], "k-."; label=@sprintf("\$t = %.3f\$", ts[end]));
35     legend(; loc=3);
36     title("Forward-Time Backward-Space, \$h = \$h\$");
37     xlabel("x");
38     ylabel("u");
39     ylim([0.0; 1.2]);
40     savefig("forward-time_backward-space_h-$.png");
41     clf();
42 end
43
44 for h in [1./10.; 1./20.; 1./40.]
45
46     println("Forward-Time Central-Space, h = $h");
47
48     xs = collect(x0:h:x1);
49     k = h * lambda;
50     ts = collect(t0:k:t1);
51     us = zeros(length(xs), length(ts));
52     us[:, 1] = map(u0, xs); # initial conditions
53
54     for n in 1:length(ts)-1
55         for m in 2:length(xs)-1
56             us[m, n+1] = us[m, n] - lambda * (us[m, n] - us[m-1, n]);
57         end
58         us[end, n+1] = us[end-1, n+1];
59     end
60
61     plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.3f\$", ts[1]));
62     plot(xs, us[:, div(length(ts), 2)], "k--"; label=@sprintf("\$t = %.3f\$",
63         ts[div(length(ts), 2)]));
64     plot(xs, us[:, end], "k-."; label=@sprintf("\$t = %.3f\$", ts[end]));
65     legend(; loc=3);
66     title("Forward-Time Central-Space, \$h = \$h\$");
67     xlabel("x");
68     ylabel("u");
69     ylim([0.0; 1.2]);
70     savefig("forward-time_central-space_h-$.png");

```

```

71     clf();
72 end
73
74 for h in [1./10.; 1./20.; 1./40.]
75
76     println("Lax-Friedrichs, h = $h");
77
78     xs = collect(x0:h:x1);
79     k = h * lambda;
80     ts = collect(t0:k:t1);
81     us = zeros(length(xs), length(ts));
82     us[:, 1] = map(u0, xs); # initial conditions
83
84     for n in 1:length(ts)-1
85         for m in 2:length(xs)-1
86             us[m, n+1] = (0.5 * (us[m+1, n] + us[m-1, n]) -
87                           0.5 * lambda * (us[m+1, n] - us[m-1, n]));
88         end
89         us[end, n+1] = us[end-1, n+1];
90     end
91
92     plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.3f\$", ts[1]));
93     plot(xs, us[:, div(length(ts), 2)], "k--"; label=@sprintf("\$t = %.3f\$",
94                                                                ts[div(length(ts), 2)]));
95     plot(xs, us[:, end], "k-."; label=@sprintf("\$t = %.3f\$", ts[end]));
96     legend(; loc=3);
97     title("Lax-Friedrichs, \$h = \$h\$");
98     xlabel("x");
99     ylabel("u");
100    ylim([0.0; 1.2]);
101    savefig("lax-friedrichs_h-$.png");
102    clf();
103 end
104
105 for h in [1./10.; 1./20.; 1./40.]
106
107     println("Leapfrog, h = $h");
108
109     xs = collect(x0:h:x1);
110     k = h * lambda;
111     ts = collect(t0:k:t1);
112     us = zeros(length(xs), length(ts));
113     us[:, 1] = map(u0, xs); # initial conditions
114
115     # one step of FTCS
116     for m in 2:length(xs)-1
117         us[m, 2] = us[m, 1] - lambda * (us[m, 1] - us[m-1, 1]);
118     end
119     us[end, 2] = us[end-1, 2];
120
121     for n in 2:length(ts)-1
122         for m in 2:length(xs)-1
123             us[m, n+1] = us[m, n-1] - lambda * (us[m+1, n] - us[m-1, n]);
124         end
125         us[end, n+1] = us[end-1, n+1];
126     end
127
128     plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.3f\$", ts[1]));
129     plot(xs, us[:, div(length(ts), 2)], "k--"; label=@sprintf("\$t = %.3f\$",
130                                                                ts[div(length(ts), 2)]));
131     plot(xs, us[:, end], "k-."; label=@sprintf("\$t = %.3f\$", ts[end]));
132     legend(; loc=3);
133     title("Leapfrog, \$h = \$h\$");
134     xlabel("x");
135     ylabel("u");
136     ylim([0.0; 1.2]);
137     savefig("leapfrog_h-$.png");
138     clf();
139 end
140

```

```

141 function total_variation(us::Vector{Float64})
142     sum = 0.0;
143     for j=2:length(us)-1
144         sum += abs(us[j+1] - us[j]);
145     end
146     return sum;
147 end
148
149 all_vars = Dict();
150 for h in [1./5.; 1./10.; 1./100.]
151
152     println("Lax-Friedrichs, h = $h");
153
154     xs = collect(-2.0:h:(15.0+0.8*h));
155     k = h * lambda;
156     ts = collect(0.0:k:(10.0+0.8*k));
157     us = zeros(length(xs), length(ts));
158     us[:, 1] = map(x -> (abs(x) <= 1) ? 2.0 : 0.0, xs); # initial conditions
159     vars = zeros(length(ts));
160     vars[1] = total_variation(us[:, 1]);
161
162     for n in 1:length(ts)-1
163         for m in 2:length(xs)-1
164             us[m, n+1] = (0.5 * (us[m+1, n] + us[m-1, n]) -
165                 0.5 * lambda * (us[m+1, n] - us[m-1, n]));
166         end
167         us[end, n+1] = us[end-1, n+1];
168         println("t = $(ts[n+1]), TV = $(total_variation(us[:, n+1]))");
169         vars[n+1] = total_variation(us[:, n+1]);
170     end
171
172     @show sum(vars) / length(vars);
173     plot(ts, vars);
174     title("Lax-Friedrichs, \h = $h\$");
175     xlabel("Time");
176     ylabel("Total Variation");
177     savefig("lax-friedrichs-TV_h-$h.png");
178     clf();
179
180     plot(xs, us[:, 1], "k-"; label=@sprintf("\$t = %.3f\$", ts[1]));
181     plot(xs, us[:, div(length(ts), 2)], "k--"; label=@sprintf("\$t = %.3f\$",
182         ts[div(length(ts), 2)]));
183     plot(xs, us[:, end], "k-."; label=@sprintf("\$t = %.3f\$", ts[end]));
184     legend(; loc=3);
185     title("Lax-Friedrichs, \h = $h\$");
186     ylim([0.0; 2.5]);
187     savefig("lax-friedrichs-c2_h-$h.png");
188     clf();
189
190     all_vars[h] = (copy(ts), copy(vars));
191 end
192
193 linetypes = ["k-", "k--", "k-."];
194 for ((h, data), linetype) in zip(all_vars, linetypes)
195     plot(data[1], data[2], linetype; label=@sprintf("\$h = %.3f\$", h));
196 end
197 legend(; loc=3);
198 xlabel("Time");
199 ylabel("Total Variation");
200 savefig("lax-friedrichs-TV_all.png");

```