

```

1  asoln(x, y, t) = exp(0.75 * t) * sin(2 * x - y) * cosh(1.5 * (x + y));
2
3  errs = [];
4  hs = [1/10; 1/20; 1/40];
5
6  for h in hs
7      k = h;
8       $\mu$  = k / (h*h);
9      println("k = $k, h = $h");
10
11     const aax = - $\mu$  / 2;
12     const bbx = ( $\mu$  + 1);
13     const ccx = aax;
14
15     const aay = - $\mu$ ;
16     const bby = (2 *  $\mu$  + 1);
17     const ccy = aay;
18
19     xs = linspace(0.0, 1.0, Int(round(1.0 / h)));
20     ys = copy(xs);
21     ts = linspace(0.0, 1.0, Int(round(1.0 / k)));
22     const M, L, K = length(xs), length(ys), length(ts);
23
24     u = zeros(M, L, K);
25     for (m, x) in zip(1:M, xs), (l, y) in zip(1:L, ys)
26         u[m, l, 1] = asoln(x, y, 0);
27     end
28
29     for n in 1:K-1
30         u_temp = zeros(M, L);
31         thalf = (ts[n]+ts[n+1]) / 2;
32         # calculate boundary terms
33         for l in 1:L
34             u_temp[l, l] = asoln(0.0, ys[l], thalf);
35             u_temp[M, l] = asoln(1.0, ys[l], thalf);
36             u[1, l, n+1] = asoln(0.0, ys[l], ts[n+1]);
37             u[M, l, n+1] = asoln(1.0, ys[l], ts[n+1]);
38         end
39         for m in 2:M-1
40             u_temp[m, 1] = asoln(xs[m], 0.0, thalf);
41             u_temp[m, L] = asoln(xs[m], 1.0, thalf);
42             u[m, 1, n+1] = asoln(xs[m], 0.0, ts[n+1]);
43             u[m, L, n+1] = asoln(xs[m], 1.0, ts[n+1]);
44         end
45
46         for l in 2:L-1
47             # calculate pi and qi for Thomas' algorithm
48             p = zeros(L);
49             q = zeros(L);
50             p[2], q[2] = 0.0, asoln(0.0, ys[l], thalf);
51             for m=2:M-1
52                 dd = u[m, l, n] + ( $\mu$  *
53                     (u[m, l+1, n] - 2 * u[m, l, n] + u[m, l-1, n]));
54                 denom = aax * p[m] + bbx;
55                 p[m+1] = -ccx / denom;
56                 q[m+1] = (dd - aax * q[m]) / denom;
57             end
58             u_temp[M, l] = asoln(1.0, ys[l], thalf);
59             for m=M-1:-1:2
60                 u_temp[m, l] = p[m+1] * u_temp[m+1, l] + q[m+1];
61             end
62             u_temp[1, l] = asoln(0.0, ys[l], thalf);
63         end
64     end

```

```

65     for m in 2:M-1
66         # calculate pi and qi for Thomas' algorithm
67         p = zeros(M);
68         q = zeros(M);
69         p[2], q[2] = 0.0, asoln(xs[m], 0.0, ts[n+1]);
70         for l=2:L-1
71             dd = u_temp[m, l] + (mu / 2 *
72                 (u_temp[m+1, l] - 2 * u_temp[m, l] + u_temp[m-1, l]));
73             denom = aay * p[l] + bby;
74             p[l+1] = -ccy / denom;
75             q[l+1] = (dd - aay * q[l]) / denom;
76         end
77         u[m, L, n+1] = asoln(xs[m], 1.0, ts[n+1]);
78         for l=L-1:-1:2
79             u[m, l, n+1] = p[l+1] * u[m, l+1, n+1] + q[l+1];
80         end
81         u[m, 1, n+1] = asoln(xs[m], 0.0, ts[n+1]);
82     end
83
84     u_exact = zeros(M, L);
85     for m in 1:M, l in 1:L
86         u_exact[m, l] = asoln(xs[m], ys[l], ts[n+1]);
87     end
88
89     if n % 5 == 0
90         println("t=$(ts[n+1]), relative L $\infty$  error: ", norm(u[:, :, n+1] - u_exact,
91             Inf) / norm(u_exact, Inf));
92         println("t=$(ts[n+1]), relative L2 error: ", norm(u[:, :, n+1] - u_exact,
93             2) / norm(u_exact, 2));
94         open(w -> begin
95             for m in 1:M, l in 1:L
96                 write(w, "$(xs[m]),$(ys[l]),$(u[m, l, n+1]),$(asoln(xs[m], ys[l], ts[n
97                     +1]))\n");
98             end
99         end, "h-$(Int(round(h*100)))_t-$(Int(round(ts[n+1]*100))).csv", "w");
100     end
101
102     u_exact = zeros(M, L);
103     for m in 1:M, l in 1:L
104         u_exact[m, l] = asoln(xs[m], ys[l], ts[K]);
105     end
106     push!(errs, maximum(map(x -> abs(x), u[:, :, K] - u_exact)));
107     println("t=1.0, relative L $\infty$  error: ", norm(u[:, :, K] - u_exact, Inf) / norm
108         (u_exact, Inf));
109     println("t=1.0, relative L2 error: ", norm(u[:, :, K] - u_exact, 2) / norm
110         (u_exact, 2));
111     println();
112
113     open(w -> begin
114         for m in 1:M, l in 1:L
115             write(w, "$(xs[m]),$(ys[l]),$(u[m, l, K]),$(asoln(xs[m], ys[l], ts[K]))
116                 \n");
117         end
118     end, "h-$(Int(round(h*100)))_end.csv", "w");
119 end
120
121 println(@sprintf("%10s %10s %10s", "h", "max(|e|)", "ratio"));
122 println(@sprintf("%10.4lf %10.4lf %10s", hs[1], errs[1], "N/A"));
123 println(@sprintf("%10.4lf %10.4lf %10lf", hs[2], errs[2], errs[1]/errs[2]));
124 println(@sprintf("%10.4lf %10.4lf %10lf", hs[3], errs[3], errs[2]/errs[3]));

```