chan up[1:PR](real edge[0:n+1]);
chan down[1:PR](real edge[0:n+1]);
chan diff(real);

process worker[w = 1 to PR] {
    int HEIGHT = n/PR;  // assume PR evenly divides n
    real grid[0:HEIGHT+1,0:n+1], new[0:HEIGHT+1,0:n+1];
    real mydiff = 0.0, otherdiff = 0.0;
    initialize grid and new, including boundaries;
    for [iters = 1 to MAXITERS by 2] {
        # compute new values for my strip
        for [i = 1 to HEIGHT, j = 1 to n]
            new[i,j] = (grid[i-1,j] + grid[i+1,j] +
                        grid[i,j-1] + grid[i,j+1]) * 0.25;
        exchange edges of new -- see text;
        # compute new values again for my strip
        for [i = 1 to HEIGHT, j = 1 to n]
            grid[i,j] = (new[i-1,j] + new[i+1,j] +
                        new[i,j-1] + new[i,j+1]) * 0.25;
        exchange edges of grid -- see text;
    }
    # compute maximum difference for my strip
    for [i = 1 to HEIGHT, j = 1 to n]
        mydiff = max(mydiff, abs(grid[i,j]-new[i,j]));
    if (w > 1)
        send diff(mydiff);
    else  // worker 1 collects differences
        for [i = 1 to w-1] {
            receive diff(otherdiff);
            mydiff = max(mydiff, otherdiff);
        }
    # maximum difference is value of mydiff in worker 1
}

Figure 11.4  Jacobi iteration using message passing.

Copyright © 2000 by Addison Wesley Longman, Inc.