real grid[0:n+1,0:n+1], new[0:n+1,0:n+1];
real maxdiff = 0.0;
initialize the grids, including the boundaries;
for [iters = 1 to MAXITERS by 2] {
    # compute new values for all interior points
    for [i = 1 to n, 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;
    # compute new values again for interior points
    for [i = 1 to n, 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;
}
# compute the maximum difference
for [i = 1 to n, j = 1 to n]
    maxdiff = max(maxdiff, abs(grid[i,j]-new[i,j]));
print the final grid and maximum difference;

Figure 11.2  Optimized sequential program for Jacobi iteration.