program main                      ! the main program
integer n, maxiters              ! common data
common /idat/ n, maxiters       ! common data
read values for n and maxiters (not shown)
call jacobi()
stop
end

subroutine jacobi()              ! implements Jacobi iteration
integer n, maxiters              ! repeat declaration of common
common /idat/ n, maxiters       ! repeat declaration of common
integer i, j, iters
integer i, j, iters
double precision grid(n,n), new(n,n)
double precision maxdiff, tempdiff
initialize grid and new (see text)

c main loop: update grids maxiters times
  do iters = 1, maxiters, 2       ! go from 1 to maxiters by 2
    do j = 2, n-1                ! update points in new
      do i = 2, n-1
        new(i,j) = (grid(i-1,j) + grid(i+1,j) +
                      grid(i,j-1) + grid(i,j+1)) * 0.25
      enddo
    enddo
  enddo
  do j = 2, n-1                  ! update points in grid
    do i = 2, n-1
      grid(i,j) = (new(i-1,j) + new(i+1,j) +
                    new(i,j-1) + new(i,j+1)) * 0.25
    enddo
  enddo
enddo

c compute maximum difference
  maxdiff = 0.0
  do j = 2, n-1
    do i = 2, n-1
      tempdiff = abs(grid(i,j)-new(i,j))
      maxdiff = max(maxdiff, tempdiff)
    enddo
  enddo
  return
end

Figure 12.3 Sequential Jacobi iteration in Fortran.