DATE: 14-02-2025 13:45:07 USER: FRTRN90 JOB: BNCHMRK **BNCHMRK** FRTRN90 FILE NAME/TYPE= TESTIN/TEST4.F90 **BNCHMRK** FRTRN90 **BNCHMRK** FRTRN90 CREATION DATE/TIME= 14-02-2025 13:45:07 **BNCHMRK** FRTRN90 **BNCHMRK** FRTRN90 FILE= 001 PAGES= 0002 LINES= 000081 **BNCHMRK** FRTRN90 SYSTEM= LINUX(6.11.2-AMD64) BNCHMRK FRTRN90 **BNCHMRK** FRTRN90 SYSID= ACID SYSUSER= ACID **BNCHMRK** FRTRN90 **BNCHMRK** FRTRN90 **BNCHMRK** FRTRN90 SYSUSER= ACID FRTRN90 BNCHMRK FORM= STANDARD FRTRN90 **BNCHMRK BNCHMRK** FRTRN90 CHAR= FONTMONO **BNCHMRK** FRTRN90 25 26 27 FFFFFFFF RRRRRRR RRRRRRR NN 9999999 00000 29 TTTTTTT 30 NN 99999999 FFFFFFFF RRRRRRR TTTTTTT RRRRRRRR NN 0000000 RR RR TT RR RR NNN NN 99 99 00 00 FF RR RR RR RR NNNN NN 99 99 00 00 TT 34 FFFFFFFF RRRRRRRR RRRRRRRR NN NN NN 99999999 00 00 TT FFFFFFFF RRRRRRR TT RRRRRRR NN NN NN 9999999 00 00 RR RR TT RR NN NNNN 99 00 00 38 FF RR RRTT RR RR NN NNN 99 00 00 FF 99999999 RR RR NN NN 0000000 RR TT RR RR RR RR RR NN 9999999 00000 42 BBBBBBBB N CCCCCCC HH HH M M RRRRRRR KKKKKKKKK 45 46 NN CCCCCCCC HH MM RRRRRRRR KKKKKKKKK BBBBBBBBB NN HH MM 47 HH MMM MMM RR BB NNN NN CC CC HH RR KKKKKKKKK BB BB NNNN NN CC HH MMMM MMMM RR RR KKKKKKKKK 49 BBBBBBBB NN NN NN CC HHHHHHHHH MM MMM MM RRRRRRRR KKKKKKKKK BBBBBBBB NN NN NN CC НННННННН MM М MM RRRRRRRR KKKKKKKKK BB NN BB NNNN CC MM MM RR RR KKKKKKKKK 53 HH HH BB BB NN MM RR RR NNN CC CC HH HH MM KKKKKKKKK BBBBBBBB NN NN CCCCCCCC HH HH MM MM RR KKKKKKKKK RR BBBBBBBB NN CCCCCCC HH HH MM MM RR RR KKKKKKKKKK 58 59 00000 00000 11 0000000 0000000 00 00 00 00 111 00 00 11 00 00 66 00 00 00 00 11 00 00 00 00 11 11 00 00 00 00 00 00 00 00 11 111111 0000000 0000000 00000 00000 111111

```
program lid_driven_cavity
      implicit none
      integer, parameter :: N = 50  ! grid size (NxN grid)
                                      ! grid spacing, time step, Reynolds number
      real :: dx, dy, dt, Re
      real :: u(N, N), v(N, N), p(N, N) ! velocity and pressure fields
      integer :: i, j, step
      real :: start_time, end_time, elapsed_time
      ! Parameters
      dx = 1.0 / (N-1)
                           ! Grid spacing in x direction
      dy = 1.0 / (N-1)
                          ! Grid spacing in y direction
                           ! Time step size
      dt = 0.001
      Re = 100
                           ! Reynolds number
                                                                                                                          18
      ! Initialize arrays
                                                                                                                          21
22
23
      u = 0.0
      v = 0.0
      p = 0.0
  18
                                                                                                                          25
26
27
      ! Initialize the top boundary (lid) velocity
      u(N, :) = 1.0
                                                                                                                          29
30
      ! Start timing
                                                                                                                          31
      call cpu_time(start_time)
                                                                                                                          33
                                                                                                                          34
35
      ! Main loop for time stepping
      do step = 1, 1000
                                                                                                                          36
                                                                                                                          37
         call compute_velocity(u, v, p, dx, dy, dt, Re)
                                                                                                                          38
         call update_pressure(p, dx, dy)
                                                                                                                          39
  30
                                                                                                                          40
                                                                                                                          41
         ! Output or check convergence
         if (mod(step, 100) == 0) then
    print *, 'Step: ', step
                                                                                                                          42
                                                                                                                          43
                                                                                                                          44
                                                                                                                          45
         end if
                                                                                                                          46
      end do
                                                                                                                          47
                                                                                                                          48
                                                                                                                          49
      ! Stop timing
      call cpu_time(end_time)
                                                                                                                          51
      elapsed_time = end_time - start_time
                                                                                                                          53
54
      print *, 'Elapsed time for CFD simulation: ', elapsed time, ' seconds'
    contains
                                                                                                                          56
                                                                                                                          57
                                                                                                                          58
59
      ! Function to update the velocity and pressure fields (simplified)
      subroutine compute_velocity(u, v, p, dx, dy, dt, Re)
                                                                                                                          61
62
        real, dimension(:,:), intent(inout) :: u, v, p
        real, intent(in) :: dx, dy, dt, Re
                                                                                                                          63
  48
        integer :: i, j
                                                                                                                          66
67
        ! Simple explicit method for velocity (simplified)
        do i = 2, N-1
                                                                                                                          68
            do i = 2, N-1
                                                                                                                          69
                                                                                                                          70
                 u(i, j) = u(i, j) - dt * ((u(i, j) * (u(i+1, j) - u(i-1, j))) / (2*dx) + &
                                              (v(i, j) * (u(i, j+1) - u(i, j-1))) / (2*dy)
                                                                                                                          73
            end do
56
        end do
                                                                                                                          76
        ! Simple velocity update for v (similar)
                                                                                                                          78
        do i = 2, N-1
            do j = 2, N-1
```

DATE: 14-02-2025 13:45:07 USER: FRTRN90 JOB: BNCHMRK PAGE: 0002

```
v(i, j) = v(i, j) - dt * ( (u(i, j) * (v(i+1, j) - v(i-1, j))) / (2*dx) + & (v(i, j) * (v(i, j+1) - v(i, j-1))) / (2*dy) )
           end do
      end do
    end subroutine compute_velocity
    ! Function to solve for pressure (simplified Poisson equation solver)
    subroutine update_pressure(p, dx, dy)
      real, dimension(:,:), intent(inout) :: p
      real, intent(in) :: dx, dy
      integer :: i, j
       ! Simple pressure Poisson equation (Jacobi iteration)
      do i = 2, N-1
           do j = 2, N-1
                p(i, j) = 0.25 * (p(i+1, j) + p(i-1, j) + p(i, j+1) + p(i, j-1))
                                                                                                                                   22
           end do
      end do
                                                                                                                                   25
26
27
    end subroutine update_pressure
 end program lid_driven_cavity
                                                                                                                                   29
30
                                                                                                                                   31
                                                                                                                                   33
                                                                                                                                   34
35
                                                                                                                                   36
                                                                                                                                   37
                                                                                                                                   38
                                                                                                                                   39
30
                                                                                                                                   40
                                                                                                                                   41
                                                                                                                                   42
                                                                                                                                   44
                                                                                                                                   45
                                                                                                                                   46
                                                                                                                                   47
                                                                                                                                   48
                                                                                                                                   49
                                                                                                                                   53
54
                                                                                                                                   57
                                                                                                                                   58
59
                                                                                                                                   63
                                                                                                                                   66
67
                                                                                                                                   70
                                                                                                                                   76
                                                                                                                                   78
```