DATE: 14-02-2025 16:25:28 USER: OPERATOR JOB: FORTRAN PAGE: 0000 FILE NAME/TYPE= STDIN FORTRAN **OPERATOR** FORTRAN **OPERATOR** FORTRAN OPERATOR CREATION DATE/TIME= 14-02-2025 16:25:28 FORTRAN **OPERATOR** 8 FORTRAN **OPERATOR** FILE= 001 PAGES= 0002 LINES= 000082 FORTRAN **OPERATOR** SYSTEM= LINUX(6.11.2-AMD64) FORTRAN OPERATOR FORTRAN **OPERATOR** SYSID= ACID SYSUSER= ACID FORTRAN **OPERATOR** FORTRAN **OPERATOR** 4 FORTRAN **OPERATOR** FORM= STANDARD FORTRAN **OPERATOR** FORTRAN **OPERATOR** CHAR= FONTMONO 17 FORTRAN OPERATOR PRT1403 VERSION= 1.2 FORTRAN **OPERATOR** 0000000 PPPPPPPP EEEEEEEE RRRRRRR AAA 0000000 RRRRRRR TTTTTTT 00000000 PPPPPPPP 00000000 RRRRRRRR EEEEEEEE RRRRRRRR AAAAA TTTTTTT 00 PP PP EE RR RR AA 00 00 RR AΑ TT 00 00 PP PP EE RR RR AA AA 00 RR RR TT 00 PPPPPPPP EEEEEEEE RRRRRRRR AA 00 00 OO RRRRRRRR TT 00 00 PPPPPPPP EEEEEEEE RRRRRRR AAAAAAAA 00 RRRRRRR TT 00 00 PP RR RR AAAAAAAA TT 00 RR RR 00 00 PP EE RR RRAA AA TT 00 00 RR RR 00000000 PP EEEEEEEE RR RR AA RR AA TT 00000000 RR EEEEEEEE RR 0000000 RR 0000000 PP RR AA AA TTRR 42 FFFFFFFF 0000000 RRRRRRR TTTTTTT RRRRRRR AAA N NN FFFFFFFF 00000000 RRRRRRRR RRRRRRRR NNTTTTTTT AAAAA NN 00 00 RR RR RR RR NNN NN TT AA AΑ 00 RR RR AA AA NNNN 00 TT RR NN FFFFFFFF 00 00 RRRRRRRR TT RRRRRRRR AA AA NN NN NN FFFFFFFF 00 00 RRRRRRR TT RRRRRRR AAAAAAAA NN 00 RR RR AAAAAAAA NN 00 TT RR FF RR 00 RR AA AA NN 00 TT RR 00000000 RR RR TT RR RR AA AA NN NN RR AA 0000000 RR TT RR AA NN N 00000 00000 11 0000000 0000000 00 00 00 00 111 11 00 00 00 00 00 00 00 00 11 11 00 00 00 00 11 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
                                                                                                                                                14
15
  dy = 1.0 / (N-1)
                      ! Grid spacing in y direction
  dt = 0.001
                       ! Time step size
                                                                                                                                                16
17
  Re = 100
                       ! Reynolds number
                                                                                                                                                18
19
  ! Initialize arrays
                                                                                                                                                21
22
23
  u = 0.0
  v = 0.0
  p = 0.0
                                                                                                                                                24
25
26
27
  ! Initialize the top boundary (lid) velocity
  u(N, :) = 1.0
                                                                                                                                                30
31
  ! Start timing
  call cpu_time(start_time)
                                                                                                                                                32
33
34
35
  ! Main loop for time stepping
  do step = 1, 1000
                                                                                                                                                36
37
     call compute_velocity(u, v, p, dx, dy, dt, Re)
     call update_pressure(p, dx, dy)
                                                                                                                                                41
      ! Output or check convergence
                                                                                                                                                42 43
     if (mod(step, 100) == 0) then
    print *, 'Step: ', step
                                                                                                                                                44 45
     end if
  end do
                                                                                                                                                49
  ! Stop timing
  call cpu_time(end_time)
  elapsed_time = end_time - start_time
                                                                                                                                                52
53
54
55
  print *, 'Elapsed time for CFD simulation: ', elapsed time,
                                                                                                                                                56
57
58
59
contains
  ! Function to update the velocity and pressure fields (simplified)
  subroutine compute_velocity(u, v, p, dx, dy, dt, Re)
    real, dimension(:,:), intent(inout) :: u, v, p
    real, intent(in) :: dx, dy, dt, Re
    integer :: i, j
                                                                                                                                                65
66
67
    ! Simple explicit method for velocity (simplified)
    do i = 2, N-1
         do i = 2, N-1
             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))
                                                                                                                                                72
73
74
75
         end do
    end do
                                                                                                                                                76
77
78
    ! Simple velocity update for v (similar)
    do i = 2, N-1
        do j = 2, N-1
```

PAGE: 0001

DATE: 14-02-2025 16:25:28 USER: OPERATOR JOB: FORTRAN 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
                                                                                                                                                                        14
15
         integer :: i, j
          ! Simple pressure Poisson equation (Jacobi iteration)
         do i = 2, N-1
              do j = 2, N-1
                                                                                                                                                                        21
22
23
24
25
26
27
                   p(i, j) = 0.25 * (p(i+1, j) + p(i-1, j) + p(i, j+1) + p(i, j-1))
              end do
         end do
       end subroutine update_pressure
    end program lid_driven_cavity
23
                                                                                                                                                                        31
32
33
34
35
                                                                                                                                                                        41
42
43
                                                                                                                                                                        43
44
45
46
47
                                                                                                                                                                        48
49
50
51
52
53
54
55
56
57
58
59
```

DATE: 14-02-2025 16:25:28 USER: OPERATOR JOB: FORTRAN PAGE: 0000 FILE NAME/TYPE= STDIN FORTRAN **OPERATOR** FORTRAN **OPERATOR** FORTRAN OPERATOR CREATION DATE/TIME= 14-02-2025 16:25:28 FORTRAN **OPERATOR** FORTRAN **OPERATOR** FILE= 002 PAGES= 0001 LINES= 000018 **FORTRAN OPERATOR** SYSTEM= LINUX(6.11.2-AMD64) FORTRAN OPERATOR FORTRAN **OPERATOR** SYSID= ACID SYSUSER= ACID FORTRAN **OPERATOR** FORTRAN **OPERATOR** 4 FORTRAN **OPERATOR** FORM= STANDARD FORTRAN **OPERATOR** FORTRAN OPERATOR CHAR= FONTMONO 17 FORTRAN OPERATOR PRT1403 VERSION= 1.2 FORTRAN **OPERATOR** 0000000 PPPPPPPP EEEEEEEE RRRRRRR AAA 0000000 RRRRRRR TTTTTTT 00000000 PPPPPPPP 00000000 RRRRRRRR EEEEEEEE RRRRRRRR AAAAA TTTTTTT 00 PP PP EE RR RR 00 00 RR AA AΑ TT 00 00 PP PP EE RR RR AA AA 00 RR RR TT 00 OO PPPPPPPP EEEEEEEE RRRRRRRR AA 00 OO RRRRRRRR TT 00 00 PPPPPPPP EEEEEEEE RRRRRRR AAAAAAAA 00 RRRRRRR TT 00 00 PP RR RR AAAAAAAA TT 00 RR RR 00 00 PP EE RR RRAA AA TT 00 00 RR RR 00000000 PP RR AA RR EEEEEEEE RR AA TT 00000000 RR 0000000 RR 0000000 PP EEEEEEEE RR RR AA AA TTRR 42 FFFFFFFF 0000000 RRRRRRR TTTTTTT RRRRRRR AAA N NN FFFFFFFF 00000000 RRRRRRRR RRRRRRRR NNAAAAA NN TTTTTTT 00 00 RR RR RR RR NNN NN TT AA AΑ 00 RR RR AA AA NNNN 00 TT RR NN FFFFFFFF 00 00 RRRRRRRR TT RRRRRRRR AA AA NN NN NN FFFFFFFF 00 00 RRRRRRR TT RRRRRRR AAAAAAAA NN 00 RR RR AAAAAAAA NN 00 TT RR FF RR 00 RR AA AA NN 00 TT RR 00000000 RR RR TT RR RR AA AA NN NN RR AA 0000000 RR TT RR AA NN N 00000 00000 222222 0000000 0000000 22222222 00 22 22 00 00 00 22 00 00 00 00 00 00 00 00 22 00 00 00 00 22 22 00 00 00 00 00 00 22 2222222 0000000 0000000 00000 00000 22222222

DATE: 14-02-2025 16:25:28

USER: OPERATOR

JOB: FORTRAN

```
program performance_test
             implicit none
             integer :: i, total
             real(8) :: start_time, end_time
             total = 0
             call cpu_time(start_time)
             do i = 1, 10000000
total = total + i
             end do
             call cpu_time(end_time)
print *, "Fortran: The sum is ", total
print *, "Fortran: Time taken = ", end_time - start_time
end program performance_test
                                                                                                                                                                                                                                     20
21
22
23
24
25
26
27
                                                                                                                                                                                                                                     28
29
30
31
32
33
34
35
36
37
38
39
40
                                                                                                                                                                                                                                     41
42
43
                                                                                                                                                                                                                                     43
44
45
46
47
                                                                                                                                                                                                                                     48
49
50
51
52
53
54
55
56
57
58
59
```

PAGE: 0001