

1
2
3
4 FORTRAN OPERATOR FILE NAME/TYPE= STDIN
5 FORTRAN OPERATOR
6 FORTRAN OPERATOR CREATION DATE/TIME= 26-02-2025 12:04:04
7 FORTRAN OPERATOR
8 FORTRAN OPERATOR FILE= 001 PAGES= 0002 LINES= 000082
9 FORTRAN OPERATOR
10 FORTRAN OPERATOR SYSTEM= LINUX(6.12.13-AMD64)
11 FORTRAN OPERATOR
12 FORTRAN OPERATOR SYSID= ACID SYSUSER= ACID
13 FORTRAN OPERATOR
14 FORTRAN OPERATOR FORM= STANDARD
15 FORTRAN OPERATOR
16 FORTRAN OPERATOR CHAR= FONTMONO
17 FORTRAN OPERATOR
18 FORTRAN OPERATOR PRT1403 VERSION= 1.3

0000000 PPPPPPPP EEEEEEEEE RRRRRRRR AAA TTTTTTTT 0000000 RRRRRRRR
0000000000 PPPPPPPP EEEEEEEEE RRRRRRRRR AAAAA TTTTTTTT 0000000000 RRRRRRRRR
00 00 PP PP EE RR RR AA AA TT 00 00 RR RR
00 00 PP PP EE RR RR AA AA TT 00 00 RR RR
00 00 PPPPPPPP EEEEEEEEE RRRRRRRRR AA AA TT 00 00 RRRRRRRRR
00 00 PPPPPPPP EEEEEEEEE RRRRRRRR AAAAAAAAAA TT 00 00 RRRRRRRR
00 00 PP EE RR RR AAAAAAAAAA TT 00 00 RR RR
00 00 PP EE RR RR AA AA TT 00 00 RR RR
0000000000 PP EEEEEEEEE RR RR AA AA TT 0000000000 RR RR
0000000 PP EEEEEEEEE RR RR AA AA TT 0000000 RR RR

FFFFFFFFF 0000000 RRRRRRRR TTTTTTTT RRRRRRRR AAA N NN
FFFFFFFFF 0000000000 RRRRRRRRR TTTTTTTT RRRRRRRRR AAAAA NN NN
FF 00 00 RR RR TT RR RR AA AA NNN NN
FF 00 00 RR RR TT RR RR AA AA NNNN NN
FFFFFFFFF 00 00 RRRRRRRRR TT RRRRRRRRR AA AA NN NN NN
FFFFFFFFF 00 00 RRRRRRRR TT RRRRRRRR AAAAAAAAAA NN NN NN
FF 00 00 RR RR TT RR RR AAAAAAAAAA NN NNNN
FF 00 00 RR RR TT RR RR AA AA NN NNN
FF 0000000000 RR RR TT RR RR AA AA NN NN
FF 0000000 RR RR TT RR RR AA AA NN N

00000 00000 1
0000000 0000000 11
00 00 00 00 111
00 00 00 00 11
00 00 00 00 11
00 00 00 00 11
00 00 00 00 11
0000000 0000000 111111
00000 00000 11111

1412THE

```
1 program lid_driven_cavity
2   implicit none
3   integer, parameter :: N = 50    ! grid size (NxN grid)
4   real :: dx, dy, dt, Re          ! grid spacing, time step, Reynolds number
5   real :: u(N, N), v(N, N), p(N, N) ! velocity and pressure fields
6   integer :: i, j, step
7   real :: start_time, end_time, elapsed_time
8
9   ! Parameters
10  dx = 1.0 / (N-1)    ! Grid spacing in x direction
11  dy = 1.0 / (N-1)    ! Grid spacing in y direction
12  dt = 0.001          ! Time step size
13  Re = 100            ! Reynolds number
14
15  ! Initialize arrays
16  u = 0.0
17  v = 0.0
18  p = 0.0
19
20  ! Initialize the top boundary (lid) velocity
21  u(N, :) = 1.0
22
23  ! Start timing
24  call cpu_time(start_time)
25
26  ! Main loop for time stepping
27  do step = 1, 1000
28    call compute_velocity(u, v, p, dx, dy, dt, Re)
29    call update_pressure(p, dx, dy)
30
31    ! Output or check convergence
32    if (mod(step, 100) == 0) then
33      print *, 'Step: ', step
34    end if
35  end do
36
37  ! Stop timing
38  call cpu_time(end_time)
39  elapsed_time = end_time - start_time
40  print *, 'Elapsed time for CFD simulation: ', elapsed_time, ' seconds'
41
42  contains
43
44  ! Function to update the velocity and pressure fields (simplified)
45  subroutine compute_velocity(u, v, p, dx, dy, dt, Re)
46    real, dimension(:, :), intent(inout) :: u, v, p
47    real, intent(in) :: dx, dy, dt, Re
48    integer :: i, j
49
50    ! Simple explicit method for velocity (simplified)
51    do i = 2, N-1
52      do j = 2, N-1
53        u(i, j) = u(i, j) - dt * ( (u(i, j) * (u(i+1, j) - u(i-1, j))) / (2*dx) + &
54                                     (v(i, j) * (u(i, j+1) - u(i, j-1))) / (2*dy) )
55      end do
56    end do
57
58    ! Simple velocity update for v (similar)
59    do i = 2, N-1
60      do j = 2, N-1
```

```
1      v(i, j) = v(i, j) - dt * ( (u(i, j) * (v(i+1, j) - v(i-1, j))) / (2*dx) + &
2                                (v(i, j) * (v(i, j+1) - v(i, j-1))) / (2*dy) )
3  end do
4  end do
5  end subroutine compute_velocity
6
7  ! Function to solve for pressure (simplified Poisson equation solver)
8  subroutine update_pressure(p, dx, dy)
9      real, dimension(:, :), intent(inout) :: p
10     real, intent(in) :: dx, dy
11     integer :: i, j
12
13     ! Simple pressure Poisson equation (Jacobi iteration)
14     do i = 2, N-1
15         do j = 2, N-1
16             p(i, j) = 0.25 * ( p(i+1, j) + p(i-1, j) + p(i, j+1) + p(i, j-1) )
17         end do
18     end do
19 end subroutine update_pressure
20
21 end program lid_driven_cavity
```

1
2
3
4 FORTRAN OPERATOR FILE NAME/TYPE= STDIN
5 FORTRAN OPERATOR
6 FORTRAN OPERATOR CREATION DATE/TIME= 26-02-2025 12:04:04
7 FORTRAN OPERATOR
8 FORTRAN OPERATOR FILE= 002 PAGES= 0001 LINES= 000018
9 FORTRAN OPERATOR
10 FORTRAN OPERATOR SYSTEM= LINUX(6.12.13-AMD64)
11 FORTRAN OPERATOR
12 FORTRAN OPERATOR SYSID= ACID SYSUSER= ACID
13 FORTRAN OPERATOR
14 FORTRAN OPERATOR FORM= STANDARD
15 FORTRAN OPERATOR
16 FORTRAN OPERATOR CHAR= FONTMONO
17 FORTRAN OPERATOR
18 FORTRAN OPERATOR PRT1403 VERSION= 1.3

19
20
21
22 0000000 PPPPPPPP EEEEEEEEE RRRRRRRR AAA TTTTTTTT 0000000 RRRRRRRR
23 000000000 PPPPPPPP EEEEEEEEE RRRRRRRRR AAAAA TTTTTTTT 000000000 RRRRRRRR
24 00 00 PP PP EE RR RR AA AA TT 00 00 RR RR
25 00 00 PP PP EE RR RR AA AA TT 00 00 RR RR
26 00 00 PPPPPPPP EEEEEEEEE RRRRRRRRR AA AA TT 00 00 RRRRRRRR
27 00 00 PPPPPPPP EEEEEEEEE RRRRRRRR AAAAAAAAA TT 00 00 RRRRRRRR
28 00 00 PP EE RR RR AAAAAAAAA TT 00 00 RR RR
29 00 00 PP EE RR RR AA AA TT 00 00 RR RR
30 000000000 PP EEEEEEEEE RR RR AA AA TT 000000000 RR RR
31 0000000 PP EEEEEEEEE RR RR AA AA TT 0000000 RR RR

32
33
34 FFFFFFFF 0000000 RRRRRRRR TTTTTTTT RRRRRRRR AAA N NN
35 FFFFFFFF 000000000 RRRRRRRRR TTTTTTTT RRRRRRRRR AAAAA NN NN
36 FF 00 00 RR RR TT RR RR AA AA NNN NN
37 FF 00 00 RR RR TT RR RR AA AA NNNN NN
38 FFFFFFFF 00 00 RRRRRRRRR TT RRRRRRRRR AA AA NN NN NN
39 FFFFFFFF 00 00 RRRRRRRR TT RRRRRRRR AAAAAAAAA NN NN NN
40 FF 00 00 RR RR TT RR RR AAAAAAAAA NN NNNN
41 FF 00 00 RR RR TT RR RR AA AA NN NNN
42 FF 000000000 RR RR TT RR RR AA AA NN NN
43 FF 0000000 RR RR TT RR RR AA AA NN N

44
45
46 00000 00000 2222222
47 0000000 0000000 222222222
48 00 00 00 00 22 22
49 00 00 00 00 22
50 00 00 00 00 22
51 00 00 00 00 22
52 00 00 00 00 22
53 00 00 00 00 22
54 0000000 0000000 22222222
55 00000 00000 22222222

```
1 program performance_test
2   implicit none
3   integer :: i, total
4   real(8) :: start_time, end_time
5
6   total = 0
7   call cpu_time(start_time)
8
9   do i = 1, 10000000
10      total = total + i
11   end do
12
13   call cpu_time(end_time)
14
15   print *, "Fortran: The sum is ", total
16   print *, "Fortran: Time taken = ", end_time - start_time
17 end program performance_test
```