3 D Poiseuille Flow

```
Clear[vx, vy, vz, Fx, Fy, Fz, p, Nx, Ny, Nz, U, ix, iy, iz, t, dt, T, s, A];
ρ = 1.2041; (*density of the fluid (air at 293.15K in this case)*)
L = 10; (*Physical length of the grid*)
a = 1; (*Physical height of the grid*)
b = 1;(*Physical depth of the grid*)
Nx = 12; (*Number of grid points over the length of the grid*)
Ny = 10;(*Number of grid points over the height of the grid*)
Nz = 10;(*Number of grid points over the depth of the grid*)
U = 1; (*Entry velocity of the fluid*)
maxtime = 400; (*The number of time-iterations*)
Rey = 100; (*Reynolds number for the fluid*)
nu = (U * a)/Rey; (*The dynamics viscosity of the fluid*)
dt = 0.02; (*The physical time between time stamps*)
Δx = L/Nx;(*The physical distance between grid points in the length direction*)
Δy = a/Ny; (*The physical distance between grid points in the height direction*)
Δz = b/Nz; (*The physical distance between grid points in the height direction*)
(*The relevant series of matrices are created*)
vx = Table[Table[Table[0, {Ny + 2}, {Nx + 1}], {Nz + 2}], {maxtime}];
vy = Table[Table[Table[0, {Ny + 1}, {Nx + 2}], {Nz + 2}], {maxtime}];
vz = Table[Table[Table[0, {Ny + 2}, {Nx + 2}], {Nz + 1}], {maxtime}];
Fx = Table[Table[Table[0, {Ny + 2}, {Nx + 1}], {Nz + 2}], {maxtime}];
Fy = Table[Table[Table[0, {Ny + 1}, {Nx + 2}], {Nz + 2}], {maxtime}];
Fz = Table[Table[Table[0, {Ny + 2}, {Nx + 2}], {Nz + 1}], {maxtime}];
p = Table[Table[Table[0, {Ny + 2}, {Nx + 2}], {Nz + 2}], {maxtime}];
s = Table[Table[Table[0, {Ny + 2}, {Nx + 2}], {Nz + 2}], {maxtime}];
A = Table[Table[Table[0, {Ny + 2}, {Nx + 2}], {Nz + 2}], {maxtime}];
(*The velocity in the length direction is initialized*)
For[l = Nz + 1, l > 1, l--,
  For[i = Nx + 1, i ≥ 1, i--,
    For[j = Ny + 1, j > 1, j--,
      vx[[1, l, j, i]] = U;
     ];
   ];
 ];
(*The A-matrix used in the pressure-subroutine with incorporated BC's (A.p=s).*)
Clear[a, n, k, A, m, w];
A = Table[0, {(Ny) * (Nx) * (Nz)}, {(Ny) * (Nx) * (Nz)}];
k = 0;
n = 0;
For[i = 1, i ≤ (Ny) * (Nx) * (Nz), i++,
  n++;
  If[n == Nx + 1, n = 1, 0];
  k++;
  If[k == Ny * Nx + 1, k = 1, 0];
```

```
For[j = 1, j ≤ (Ny) * (Nx) * (Nz), j++,
 A[[i, j]] = (-2 (1/((Δy)^2) + 1/((Δx)^2) + 1/((Δz)^2))
      + If[n == 1, 1/((Δx)^2), 0]
      + If[n == Nx, 1/((Δx)^2), 0]
      + If[1 ≤ k ≤ (Ny - 1) * Nx, 0, 1/((Δy)^2)]
      + If[Nx + 1 ≤ k ≤ Ny * Nx, 0, 1/((Δy)^2)]
      + If[1 ≤ i ≤ Nx * Ny, 1/((Δz)^2), 0]
      + If[Ny * Nx * Nz - Nx * Ny + 1 ≤ i ≤ Ny * Nx * Nz, 1/((Δz)^2), 0]) *
    KroneckerDelta[i, j] +

   1/((Δx)^2) *
    (If[n ≠ Nx, KroneckerDelta[i, j - 1], 0] + If[n ≠ 1, KroneckerDelta[i, j + 1], 0]) +

   1/((Δy)^2) * (If[1 ≤ k ≤ (Ny - 1) * Nx, KroneckerDelta[i, j - (Nx)], 0] +
      If[Nx + 1 ≤ k ≤ Ny * Nx, KroneckerDelta[i, j + Nx], 0]) +

   1/((Δz)^2) * (KroneckerDelta[i, j - (Nx * Ny)] + KroneckerDelta[i, j + Nx * Ny])
 ];
];
(*The correct solution is only obtained
 when the rank of A is equal to the number of rows*)
Clear[n];
n = 0;
For[i = 1, i ≤ (Ny) * (Nx) * (Nz), i++,
  n++;
  If[Dimensions[A][[1]] ≠ MatrixRank[A],
   A = A[[1 ;; (Ny) * (Nx) * (Nz) - n, 1 ;; (Ny) * (Nx) * (Nz) - n]];
   , Indeterminate];
 ];

(*Begin the time loop untill maxtime-1*)
For[t = 1, t < maxtime, t++,
  (*calculate the next F_x-vlaue for internal points*)
  For[iz = 2, iz ≤ Nz + 1, iz++,
   (*iz runs over the depth physical space + dummy boundary*)
   For[iy = 2, iy ≤ Ny + 1, iy++,
     (*iy runs over the vertical physical space + dummy boundary*)
     For[ix = 2, ix ≤ Nx, ix++, (*ix runs over the horizontal
         physical space+ dummy boundary*)

       Fx[[t, iz, iy, ix]] =
```

```
          -ρ * (vx[[t, iz, iy, ix]] * (vx[[t, iz, iy, ix + 1]] - vx[[t, iz, iy, ix - 1]]) / (2 * Δx)
              + 1/4 (vy[[t, iz, iy, ix]] + vy[[t, iz, iy, ix + 1]] +
                 vy[[t, iz, iy - 1, ix]] + vy[[t, iz, iy - 1, ix + 1]]) *
                (vx[[t, iz, iy + 1, ix]] - vx[[t, iz, iy - 1, ix]]) / (2 * Δy)
              + 1/4 (vz[[t, iz, iy, ix]] + vz[[t, iz, iy + 1, ix]] +
                 vz[[t, iz - 1, iy, ix]] + vz[[t, iz - 1, iy + 1, ix]]) *
                (vx[[t, iz + 1, iy, ix]] - vx[[t, iz - 1, iy, ix]]) / (2 * Δz))
          + nu * (1/(Δx)^2 (vx[[t, iz, iy, ix + 1]] + vx[[t, iz, iy, ix - 1]] -
                 2 vx[[t, iz, iy, ix]])
              + 1/(Δy)^2 (vx[[t, iz, iy + 1, ix]] + vx[[t, iz, iy - 1, ix]] -
                 2 vx[[t, iz, iy, ix]])
              + 1/(Δz)^2 (vx[[t, iz + 1, iy, ix]] + vx[[t, iz - 1, iy, ix]] -
                 2 vx[[t, iz, iy, ix]]));

      ];
    ];
  ];

(*calculate the next F_y-vlaue for internal points*)
For[iy = 2, iy ≤ Ny, iy++,
 For[ix = 2, ix ≤ Nx + 1, ix++,
   For[iz = 2, iz ≤ Nz + 1, iz++,
     Fy[[t, iz, iy, ix]] = -ρ *
        (1/4 (vx[[t, iz, iy, ix]] + vx[[t, iz, iy, ix - 1]] + vx[[t, iz, iy + 1, ix]] + vx[[
               t, iz, iy + 1, ix - 1]]) * (vy[[t, iz, iy, ix + 1]] - vy[[t, iz, iy, ix - 1]]) / (2 * Δx)
           vy[[t, iz, iy, ix]] * (vy[[t, iz, iy + 1, ix]] - vy[[t, iz, iy - 1, ix]]) / (2 * Δy)
          + 1/4 (vz[[t, iz, iy, ix]] + vz[[t, iz - 1, iy, ix]] +
             vz[[t, iz, iy + 1, ix]] + vz[[t, iz - 1, iy + 1, ix]]) *
            (vy[[t, iz + 1, iy, ix]] - vy[[t, iz - 1, iy, ix]]) / (2 * Δz))
      + nu * (1/(Δx)^2 (vy[[t, iz, iy, ix + 1]] + vy[[t, iz, iy, ix - 1]] -
             2 vy[[t, iz, iy, ix]])
          + 1/(Δy)^2 (vy[[t, iz, iy + 1, ix]] + vy[[t, iz, iy - 1, ix]] -
             2 vy[[t, iz, iy, ix]])
          + 1/(Δz)^2 (vy[[t, iz + 1, iy, ix]] + vy[[t, iz - 1, iy, ix]] -
```

```
                2 vy[[t, iz, iy, ix]]));
      ];
    ];
  ];

  (*calculate the next F_z-vlaue for internal points*)
  For[iy = 2, iy ≤ Ny + 1, iy++,
   For[ix = 2, ix ≤ Nx + 1, ix++,
     For[iz = 2, iz ≤ Nz, iz++,
       Fz[[t, iz, iy, ix]] = -ρ *
          (1/4 (vx[[t, iz, iy, ix]] + vx[[t, iz, iy, ix - 1]] + vx[[t, iz + 1, iy, ix]] + vx[[
                t, iz + 1, iy, ix - 1]]) * (vz[[t, iz, iy, ix + 1]] - vz[[t, iz, iy, ix - 1]])/(2 * Δx)
            + 1/4 (vy[[t, iz, iy, ix]] + vy[[t, iz + 1, iy, ix]] +
               vy[[t, iz, iy - 1, ix]] + vy[[t, iz + 1, iy - 1, ix]]) *
             (vz[[t, iz, iy + 1, ix]] - vz[[t, iz, iy - 1, ix]])/(2 * Δy)
            + vz[[t, iz, iy, ix]] * (vz[[t, iz + 1, iy, ix]] - vz[[t, iz - 1, iy, ix]])/(2 * Δz))
         + nu * (1/(Δx) ^2
             (vz[[t, iz, iy, ix + 1]] + vz[[t, iz, iy, ix - 1]] - 2 vz[[t, iz, iy, ix]])
            + 1/(Δy) ^2 (vz[[t, iz, iy + 1, ix]] + vz[[t, iz, iy - 1, ix]] -
               2 vz[[t, iz, iy, ix]])
            + 1/(Δz) ^2 (vz[[t, iz + 1, iy, ix]] + vz[[t, iz - 1, iy, ix]] -
               2 vz[[t, iz, iy, ix]]));
      ];
    ];
  ];

  (*calculate the next source-vlaue for internal points*)
  For[ix = 2, ix ≤ Nx + 1, ix++,
   For[iy = 2, iy ≤ Ny + 1, iy++,
     For[iz = 2, iz ≤ Nz + 1, iz++,
       s[[t, iz, iy, ix]] = (Fx[[t, iz, iy, ix]] - Fx[[t, iz, iy, ix - 1]])/Δx +
          (Fy[[t, iz, iy, ix]] - Fy[[t, iz, iy - 1, ix]])/Δy +
          (Fz[[t, iz, iy, ix]] - Fz[[t, iz - 1, iy, ix]])/Δz +
          ρ/dt * ((vx[[t, iz, iy, ix]] - vx[[t, iz, iy, ix - 1]])/Δx +
             (vy[[t, iz, iy, ix]] - vy[[t, iz, iy - 1, ix]])/Δy +
```

```
            vz[[t, iz, iy, ix]] - vz[[t, iz - 1, iy, ix]]
            ─────────────────────────────────────────────);
                              Δz
    ];
  ];
];

(*Start the subroutine that determine the pressure corresponding to the source*)
(*First, arrange the s-matrix indices in a vector, S*)
Clear[S, n, i, j, k];
S = Table[0, {Nx * Ny * Nz}];
n = 0;
For[k = 2, k ≤ Nz + 1, k++,
 For[i = 2, i ≤ Ny + 1, i++,
   For[j = 2, j ≤ Nx + 1, j++,
     n++;
     S[[n]] = s[[t, k, i, j]];
    ];
  ];
];
(*Next, remove entries from the end of S corresponding
  to the rows removed in A in order to obtain rank(A)=rows(A)*)
Clear[temp];
S = S[[1 ;; Dimensions[A][[1]]]];
(*Solve A.x=S. Inverting such large matrices is computationally too heavy,
so approximate methods must be applied. This can be done via SOR,
but the inbuilt function is much, much faster.*)
temp = LinearSolve[A, S];
(*"temp" is a vector corresponding to S. This is rearranged
 into a matrix with pressure measurements, corresponding to s*)
n = 0;
For[k = 1, k ≤ Nz, k++,
 For[i = 1, i ≤ Ny, i++,
   For[j = 1, j ≤ Nx, j++,
     n++;
     If[n ≤ Dimensions[temp][[1]] ⋀ temp[[j + (i - 1) * Nx + (k - 1) * Ny * Nx]] > 10^-8,
      p[[t, k + 1, i + 1, j + 1]] = temp[[j + (i - 1) * Nx + (k - 1) * Ny * Nx]];
      , Indeterminate];
    ];
  ];
];
(*The boundary terms are set equal to the outer interla points. Without
  incorporating the BC's in the A-matrix this would eb wrong*)
For[iz = 2, iz ≤ Nz + 1, iz++,
 For[ix = 2, ix ≤ Nx + 1, ix++,
   p[[t, iz, 1, ix]] = p[[t, iz, 2, ix]];
   p[[t, iz, Ny + 2, ix]] = p[[t, iz, Ny + 1, ix]];
  ];
];
For[iz = 2, iz ≤ Nz + 1, iz++,
 For[iy = 2, iy ≤ Ny + 1, iy++,
   p[[t, iz, iy, 1]] = p[[t, iz, iy, 2]];
   p[[t, iz, iy, Nx + 2]] = p[[t, iz, iy, Nx + 1]];
  ];
];
```

```
For[ix = 2, ix ≤ Nx + 1, ix++,
 For[iy = 2, iy ≤ Ny + 1, iy++,
   p[[t, 1, iy, ix]] = p[[t, 2, iy, ix]];
   p[[t, Nz + 2, iy, ix]] = p[[t, Nz + 1, iy, ix]];
  ];
];
(*calculate the next v_x-vlaue for internal points*)
For[iz = 1, iz ≤ Nz + 2, iz++,
 For[ix = 1, ix ≤ Nx + 2, ix++,
   For[iy = 1, iy ≤ Ny + 2, iy++,

     If[2 ≤ iz ≤ Nz + 1,
      If[2 ≤ iy ≤ Ny + 1,
        If[2 ≤ ix ≤ Nx,
          vx[[t + 1, iz, iy, ix]] = vx[[t, iz, iy, ix]] + dt/ρ *
               (Fx[[t, iz, iy, ix]] - (p[[t, iz, iy, ix + 1]] - p[[t, iz, iy, ix]])/Δx);
          , Indeterminate];
        , Indeterminate];
      , Indeterminate];
     (*calculate the next v_y-vlaue for internal points*)
     If[2 ≤ iz ≤ Nz + 1,
      If[2 ≤ iy ≤ Ny,
        If[2 ≤ ix ≤ Nx + 1,
          vy[[t + 1, iz, iy, ix]] = vy[[t, iz, iy, ix]] + dt/ρ *
               (Fy[[t, iz, iy, ix]] - (p[[t, iz, iy + 1, ix]] - p[[t, iz, iy, ix]])/Δy);
          , Indeterminate];
        , Indeterminate];
      , Indeterminate];
     (*calculate the next v_z-vlaue for internal points*)
     If[2 ≤ iz ≤ Nz,
      If[2 ≤ iy ≤ Ny + 1,
        If[2 ≤ ix ≤ Nx + 1,
          vz[[t + 1, iz, iy, ix]] = vz[[t, iz, iy, ix]] + dt/ρ *
               (Fz[[t, iz, iy, ix]] - (p[[t, iz + 1, iy, ix]] - p[[t, iz, iy, ix]])/Δz);
          , Indeterminate];
        , Indeterminate];
      , Indeterminate];
     (*Applty BC's for v_x, v_y and v_z*)

     If[2 ≤ iz ≤ Nz + 1,
      If[2 ≤ iy ≤ Ny + 1,
        vx[[t + 1, iz, iy, 1]] = U;
        vx[[t + 1, iz, iy, Nx + 1]] = vx[[t + 1, iz, iy, Nx]];
        , Indeterminate];
      , Indeterminate];
```

```
If[2 ≤ iz ≤ Nz + 1,
 If[1 ≤ ix ≤ Nx + 1,
   vx[[t + 1, iz, 1, ix]] = -vx[[t + 1, iz, 2, ix]];
   vx[[t + 1, iz, Ny + 2, ix]] = -vx[[t + 1, iz, Ny + 1, ix]];
   , Indeterminate];
 , Indeterminate];

If[2 ≤ iy ≤ Ny + 1,
 If[1 ≤ ix ≤ Nx + 1,
   vx[[t + 1, 1, iy, ix]] = -vx[[t + 1, 2, iy, ix]];
   vx[[t + 1, Nz + 2, iy, ix]] = -vx[[t + 1, Nz + 1, iy, ix]];
   , Indeterminate];
 , Indeterminate];

If[2 ≤ iz ≤ Nz + 1,
 If[2 ≤ iy ≤ Ny,
   vy[[t + 1, iz, iy, 1]] = -vy[[t + 1, iz, iy, 2]];
   vy[[t + 1, iz, iy, Nx + 2]] = vy[[t + 1, iz, iy, Nx + 1]];
   , Indeterminate];
 , Indeterminate];

If[2 ≤ iz ≤ Nz + 1,
 If[1 ≤ ix ≤ Nx + 2,
   vy[[t + 1, iz, 1, ix]] = 0;
   vy[[t + 1, iz, Ny + 1, ix]] = 0;
   , Indeterminate];
 , Indeterminate];

If[2 ≤ iy ≤ Ny + 1,
 If[1 ≤ ix ≤ Nx + 2,
   vy[[t + 1, 1, iy, ix]] = -vy[[t + 1, 2, iy, ix]];
   vy[[t + 1, Nz + 2, iy, ix]] = -vy[[t + 1, Nz + 1, iy, ix]];
   , Indeterminate];
 , Indeterminate];

If[2 ≤ iz ≤ Nz + 1,
 If[2 ≤ iy ≤ Ny,
   vz[[t + 1, iz, iy, 1]] = -vz[[t + 1, iz, iy, 2]];
   vz[[t + 1, iz, iy, Nx + 2]] = vz[[t + 1, iz, iy, 1]];
   , Indeterminate];
 , Indeterminate];

If[2 ≤ iy ≤ Ny + 1,
 If[1 ≤ ix ≤ Nx + 2,
   vz[[t + 1, 1, iy, ix]] = 0;
   vz[[t + 1, Nz + 1, iy, ix]] = 0;
   , Indeterminate];
 , Indeterminate];

If[1 ≤ iz ≤ Ny + 1,
 If[1 ≤ ix ≤ Nx + 2,
   vz[[t + 1, iz, 1, ix]] = -vz[[t + 1, iz, 2, ix]];
   vz[[t + 1, iz, Ny + 2, ix]] = -vz[[t + 1, iz, Ny + 1, ix]];
   , Indeterminate];
```

```
            , Indeterminate];
          ];
        ];
      ];
    ];
```