Direct Numerical Simulations of Multiphase Flows-3

A Simple Solver for Variable Density Flow (2 of 3)

Gretar Tryggvason

1. Here we continue to develop a simple solver for flows with variable density.

DNS of Multiphase Flows

Corrected Velocities

2. Once the predicted velocities have been found, we need to look at how to evaluate the corrected velocities.

DNS of Multiphase Flows

Discretization in space—Corrector step

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \frac{\nabla_h P}{\rho^{n+1}}$$

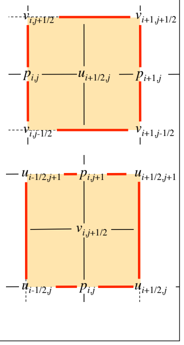
In component form:

$$u^{n+1}_{i+1/2,j} = u^*_{i+1/2,j} - \frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i+1,j} + \rho^{n+1}_{i,j})} \frac{p_{i+1,j} - p_{i,j}}{\Delta x}$$

$$v^{n+1}_{i,j+1/2} = v^*_{i,j+1/2} - \frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j+1} + \rho^{n+1}_{i,j})} \frac{p_{i,j+1} - p_{i,j}}{\Delta y}$$

Use linear interpolation where quantities are not defined

$$\rho^{n+1}_{i+1/2,j} = \frac{1}{2}(\rho^{n+1}_{i+1,j} + \rho^{n+1}_{i,j})$$



3. The incompressible velocity at the new time, u n+1, is the predicted velocity, u star, minus delta t over rho times the pressure gradient needed to make the new velocity divergence free. For the horizontal velocity at i+1/2,j we use the difference between the pressures on the right boundary of the velocity control volume and the pressure on the left boundary. These are given directly as the pressures at the center of the pressure control volume to the right, p i+1,j and the pressure at the center of the of the pressure control volume to the left, p i,j. The density is needed at the center of the velocity control volume and, since it is assumed to be known at the center of the pressure control volume, we interpolate by taking the average. Similarly, the vertical velocity at I,j+1/2 is updated by adding the difference between the pressures at I,j+1 and I,j, divided by the averages of the densities at the same points.

The pressure equation

4. In writing the equation for the correction velocities we assumed that the pressures are known. This is, of course, not the case and before we do the update, we need to find the pressure.

To correct the predicted velocity, we need the pressure. The pressure equation is derived by substituting the expression for the correction velocities into the mass conservation equation. Once the pressure has been found, we can use it to correct the predicted velocity.

Schematically

$$\left.\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \frac{\nabla_h p}{\rho^{n+1}} \\ \nabla_h \cdot \mathbf{u}^{n+1} &= 0 \end{aligned}\right\} \Rightarrow \nabla_h \cdot \left(\frac{1}{\rho^{n+1}} \nabla_h p \right) = \frac{1}{\Delta t} \nabla_h \cdot \mathbf{u}^*$$

The detailed equations are derived by examining the discrete approximations directly:

5. Although the primitive, or the velocity-pressure, form of the Navier Stokes equations does not have a separate equation for the pressure, we can derive it by combining the incompressibility conditions with the momentum equations. When we use the projection method for the time integration and split the momentum equation into a prediction and a correction step, the pressure appears in the correction step. We don't know the velocity at n+1, but the incompressibility equation says that its divergence should be zero. Thus, taking the divergence of the correction equation eliminates the new unknown velocity and leaves us with a relationship between the pressure and the predicted velocity, which we know. We generally derive the pressure equation by working directly with the discrete approximations and that is what we will go through that in the next few slides.

To derive an equation for the pressure, start with the conservation of mass equation

$$\frac{u^{n+1}_{i+1/2,j} - u^{n+1}_{i-1/2,j}}{\Delta x} + \frac{v^{n+1}_{i,j+1/2} - v^{n+1}_{i,j-1/2}}{\Delta y} = 0$$

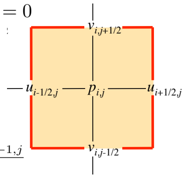
and substitute the corrected velocities

$$u^{n+1}_{i+1/2,j} = u^*_{i+1/2,j} - \frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i+1,j} + \rho^{n+1}_{i,j})} \frac{p_{i+1,j} - p_{i,j}}{\Delta x}$$

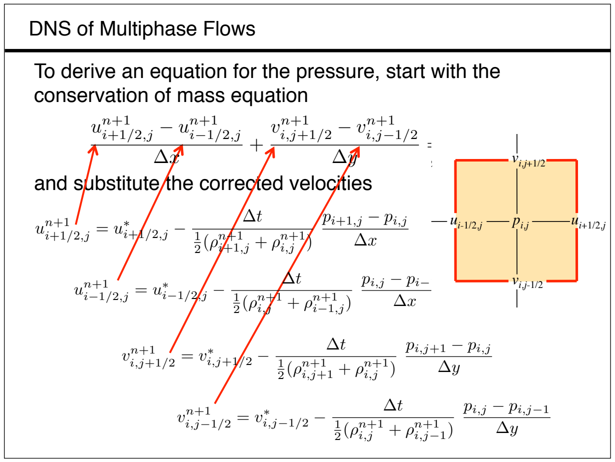
$$u^{n+1}_{i-1/2,j} = u^*_{i-1/2,j} - \frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j} + \rho^{n+1}_{i-1,j})} \frac{p_{i,j} - p_{i-1,j}}{\Delta x}$$

$$v^{n+1}_{i,j+1/2} = v^*_{i,j+1/2} - \frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j+1} + \rho^{n+1}_{i,j})} \frac{p_{i,j+1} - p_{i,j}}{\Delta y}$$

$$v^{n+1}_{i,j-1/2} = v^*_{i,j-1/2} - \frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j} + \rho^{n+1}_{i,j-1})} \frac{p_{i,j} - p_{i,j-1}}{\Delta y}$$



6. To derive an equation for the pressure at point i,j, we use the incompressibility condition for a control volume around that point, written at the top of the slide, which states that the inflow and the outflow must be equal. The correction equations for the normal velocity on each side, written below, relates the new normal velocity to the predicted velocity and the differences in pressure at i,j and the outside pressure. Thus, u n+1 at i+1/2,j is equal to the predicted velocity at the same point plus the differences between the pressure at i+1,j and i,j, times delta t, divided by delta x and the density at i,j. The other velocities are given by similar expressions.



DNS of Multiphase Flows

To derive an equation for the pressure, start with the conservation of mass equation

$$\frac{u^{n+1}_{i+1/2,j}-u^{n+1}_{i-1/2,j}}{\Delta x}+\frac{v^{n+1}_{i,j+1/2}-v^{n+1}_{i,j-1/2}}{\Delta y} =$$

and substitute the corrected velocities

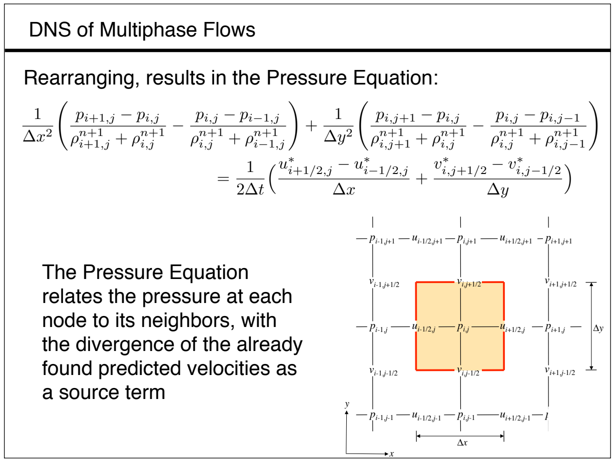
$$u^{n+1}_{i+1/2,j}=u^*_{i+1/2,j}-\frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i+1,j}+\rho^{n+1}_{i,j})}\frac{p_{i+1,j}-p_{i,j}}{\Delta x}$$

$$u^{n+1}_{i-1/2,j}=u^*_{i-1/2,j}-\frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j}+\rho^{n+1}_{i-1,j})}\frac{p_{i,j}-p_{i-}}{\Delta x}$$

$$v^{n+1}_{i,j+1/2}=v^*_{i,j+1/2}-\frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j+1}+\rho^{n+1}_{i,j})}\frac{p_{i,j+1}-p_{i,j}}{\Delta y}$$

$$v^{n+1}_{i,j-1/2}=v^*_{i,j-1/2}-\frac{\Delta t}{\frac{1}{2}(\rho^{n+1}_{i,j}+\rho^{n+1}_{i,j-1})}\frac{p_{i,j}-p_{i,j-1}}{\Delta y}$$

7. Substituting the expressions for the correction velocities on each side of the control volume into the incompressibility equation eliminates the unknown velocities at the new time level and gives us an equation for the pressures.



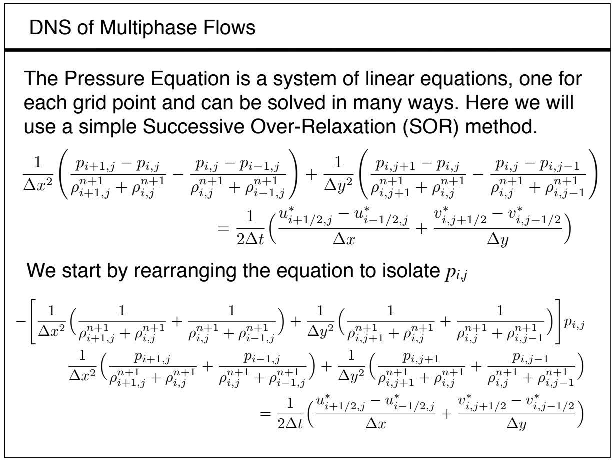
DNS of Multiphase Flows

Rearranging, results in the Pressure Equation:

$$\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}-p_{i,j}}{\rho^{n+1}_{i+1,j}+\rho^{n+1}_{i,j}}-\frac{p_{i,j}-p_{i-1,j}}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i-1,j}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}-p_{i,j}}{\rho^{n+1}_{i,j+1}+\rho^{n+1}_{i,j}}-\frac{p_{i,j}-p_{i,j-1}}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i,j-1}}\right)$$
$$=\frac{1}{2\Delta t}\left(\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y}\right)$$

The Pressure Equation relates the pressure at each node to its neighbors, with the divergence of the already found predicted velocities as a source term

8. The result is a set of linear equations that relate the pressure at each point to the pressure at the neighboring points and the predicted velocities, which are known. Here we have rearranged the equation slightly so the known quantities, namely the predicted velocities u* and v* appear on the RHS. Since we started by advecting the densities, the rhos at n+1 are also known and the coefficients multiplying the pressure on the LHS can be computed.



DNS of Multiphase Flows

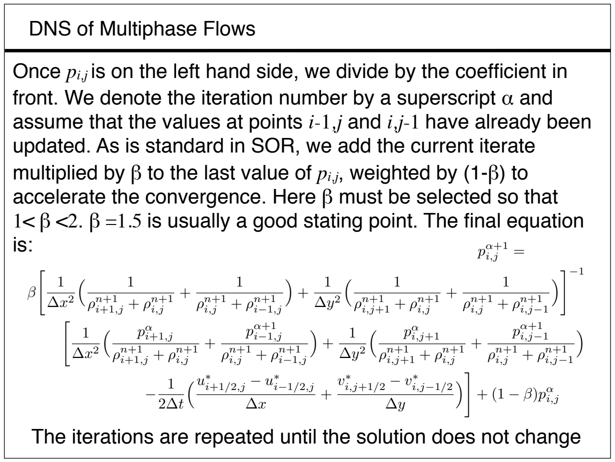
The Pressure Equation is a system of linear equations, one for each grid point and can be solved in many ways. Here we will use a simple Successive Over-Relaxation (SOR) method.

$$\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}-p_{i,j}}{\rho^{n+1}_{i+1,j}+\rho^{n+1}_{i,j}}-\frac{p_{i,j}-p_{i-1,j}}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i-1,j}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}-p_{i,j}}{\rho^{n+1}_{i,j+1}+\rho^{n+1}_{i,j}}-\frac{p_{i,j}-p_{i,j-1}}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i,j-1}}\right)$$
$$=\frac{1}{2\Delta t}\left(\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y}\right)$$

We start by rearranging the equation to isolate p_{ij}

$$-\left[\frac{1}{\Delta x^2}\left(\frac{1}{\rho^{n+1}_{i+1,j}+\rho^{n+1}_{i,j}}+\frac{1}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i-1,j}}\right)+\frac{1}{\Delta y^2}\left(\frac{1}{\rho^{n+1}_{i,j+1}+\rho^{n+1}_{i,j}}+\frac{1}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i,j-1}}\right)\right]p_{i,j}$$
$$\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}}{\rho^{n+1}_{i+1,j}+\rho^{n+1}_{i,j}}+\frac{p_{i-1,j}}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i-1,j}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}}{\rho^{n+1}_{i,j+1}+\rho^{n+1}_{i,j}}+\frac{p_{i,j-1}}{\rho^{n+1}_{i,j}+\rho^{n+1}_{i,j-1}}\right)$$
$$=\frac{1}{2\Delta t}\left(\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y}\right)$$

9. We have one pressure equation for each grid point and since they link the pressure points to the neighboring pressure points we need to solve a system of linear equations. There are many, many ways to do so and, indeed, solving linear equations is a very large and very advanced field. Here, however, we will solve those equations in the simplest way possible, using an elementary successive over relaxation or SOR method. We start by rearranging the equation for p at point i,j to isolate it. Here we have gathered everything multiplying p i,j into the square brackets.



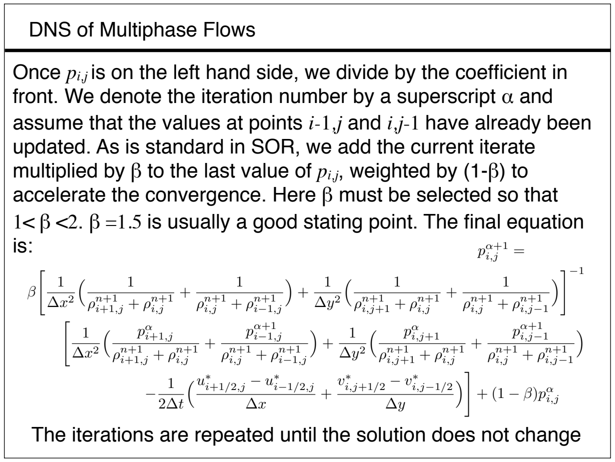
DNS of Multiphase Flows

Once $p_{i,j}$ is on the left hand side, we divide by the coefficient in front. We denote the iteration number by a superscript α and assume that the values at points i-1,j and i,j-1 have already been updated. As is standard in SOR, we add the current iterate multiplied by β to the last value of $p_{i,j}$, weighted by (1-β) to accelerate the convergence. Here β must be selected so that $1<\beta<2$. $\beta=1.5$ is usually a good stating point. The final equation is:

$$p_{i,j}^{\alpha+1} = \beta\left[\frac{1}{\Delta x^2}\left(\frac{1}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{1}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right)\right]^{-1} \left[\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}^{\alpha}}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i-1,j}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}^{\alpha}}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i,j-1}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right) -\frac{1}{2\Delta t}\left(\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y}\right)\right]+(1-\beta)p^\alpha_{i,j}$$

The iterations are repeated until the solution does not change

10-1. At each step the new densities, rho n+1 at each grid points are known, so we can compute the constant multiplying p i,j and divide by it. Putting everything else on the left hand side gives us an equation for p i,j. The problem is, of course, that the pressures at i+1,j and so on are unknown. Thus, we solve the equation by iteration where we start by guessing the pressure. Then we use our equation to compute new pressures that are then used to compute the pressure again and so on, until the pressures do not change. Fortunately this process generally converges, although we may need a large number of iterations. Since the pressure values converge smoothly, and it is clear what the trend are, we can sometimes accelerate the process by extrapolating the new value based on how different it is from the old value. This is easily done by multiplying the new value with a constant that is larger than one and adding the old value multiplied by one minus this constant. Here we denote the constant by beta and note that it can be shown that beta must be smaller than two for the results to converge.



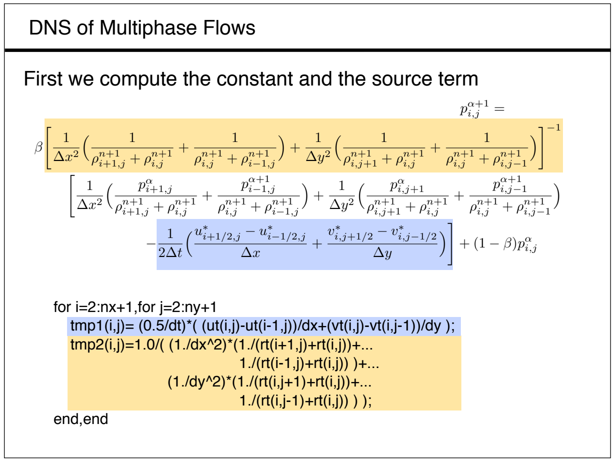
DNS of Multiphase Flows

Once $p_{i,j}$ is on the left hand side, we divide by the coefficient in front. We denote the iteration number by a superscript α and assume that the values at points i-1,j and i,j-1 have already been updated. As is standard in SOR, we add the current iterate multiplied by β to the last value of $p_{i,j}$, weighted by (1-β) to accelerate the convergence. Here β must be selected so that $1<\beta<2$. $\beta=1.5$ is usually a good stating point. The final equation is:

$$p_{i,j}^{\alpha+1} = \beta\left[\frac{1}{\Delta x^2}\left(\frac{1}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{1}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right)\right]^{-1} \left[\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}^{\alpha}}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i-1,j}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}^{\alpha}}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i,j-1}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right) -\frac{1}{2\Delta t}\left(\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y}\right)\right]+(1-\beta)p^\alpha_{i,j}$$

The iterations are repeated until the solution does not change

10-2. Thus beta must be larger than one and smaller than 2 so beta equal to 1.5 is usually a reasonable start. Larger beta gives faster convergence when it works but larger beta can also cause the iteration to divergence.



DNS of Multiphase Flows

First we compute the constant and the source term

$$p_{i,j}^{\alpha+1} = \beta\left[\frac{1}{\Delta x^2}\left(\frac{1}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{1}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right)\right]^{-1} \left[\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}^{\alpha}}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i-1,j}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}^{\alpha}}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i,j-1}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right) -\frac{1}{2\Delta t}\left(\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y}\right)\right]+(1-\beta)p^\alpha_{i,j}$$

```
for i=2:nx+1,for j=2:ny+1
   tmp1(i,j)= (0.5/dt)*( (ut(i,j)-ut(i-1,j))/dx+(vt(i,j)-vt(i,j-1))/dy );
   tmp2(i,j)=1.0/( (1./dx^2)*(1./(rt(i+1,j)+rt(i,j))+...
                                  1./(rt(i-1,j)+rt(i,j)) )+...
                        (1./dy^2)*(1./(rt(i,j+1)+rt(i,j))+...
                                  1./(rt(i,j-1)+rt(i,j)) ) );
end,end
```

11. During the iteration we re-compute the equation over and over again as the pressures change. However, the densities and the source term do not change so it makes sense to compute the coefficient in front of p i,j (highlighted by yellow) and the source term (purple) at the beginning and store those. Thus, we put the following code snippet before the pressure iteration. Although we do not need to compute the coefficient and the source, for each iteration, we generally have to compute them again for each time step since both the densities and the predicted velocities change as the solution evolves in time. Notice that we only solve for the pressures inside the computational domain so i goes from 2 to nx+1 and j goes from 2 to ny+1.

DNS of Multiphase Flows

$$p_{i,j}^{\alpha+1} =$$

$$\beta\left[\frac{1}{\Delta x^2}\left(\frac{1}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{1}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{1}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right)\right]^{-1}$$

$$\left[\frac{1}{\Delta x^2}\left(\frac{p_{i+1,j}^{\alpha}}{\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i-1,j}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i-1,j}^{n+1}}\right)+\frac{1}{\Delta y^2}\left(\frac{p_{i,j+1}^{\alpha}}{\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1}}+\frac{p_{i,j-1}^{\alpha+1}}{\rho_{i,j}^{n+1}+\rho_{i,j-1}^{n+1}}\right)\right.$$

$$\left.-\frac{1}{2\Delta t}\left(\frac{u_{i+1/2,j}^{*}-u_{i-1/2,j}^{*}}{\Delta x}+\frac{v_{i,j+1/2}^{*}-v_{i,j-1/2}^{*}}{\Delta y}\right)\right]+(1-\beta)p_{i,j}^{\alpha}$$

```
for it=1:maxit                    % SOLVE FOR PRESSURE
  oldArray=p;
  for i=2:nx+1,for j=2:ny+1
     p(i,j)=(1.0-beta)*p(i,j)+beta* tmp2(i,j)*(...
     (1./dx^2)*( p(i+1,j)/(rt(i+1,j)+rt(i,j))+...
                 p(i-1,j)/(rt(i-1,j)+rt(i,j)) )+...
     (1./dy^2)*( p(i,j+1)/(rt(i,j+1)+rt(i,j))+...
                 p(i,j-1)/(rt(i,j-1)+rt(i,j)) ) - tmp1(i,j));
  end,end
  if max(max(abs(oldArray-p))) <maxError, break,end
end
```

The coefficient arrays tmp1 and tmp2 are computed at the beginning of the iteration and do not need to be updated

12. Once the coefficients and the source has been computed, we iterate to find the pressure. Notice that we only need one pressure array since the new values are used as soon as they have been computed. To stop the iteration we simply compare the new pressure with the old one and if the change is sufficiently small we declare victory and accept the pressure values as the correct answer. This is not a particularly sophisticated way to evaluate when the iteration has converged but it is simple and works sufficiently well for our purpose.

DNS of Multiphase Flows

Correct the velocity

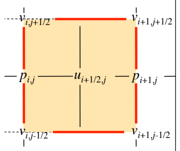
13. Once the pressures have been found, we can correct the predicted velocities by adding the discrete pressure gradient.

DNS of Multiphase Flows

After the pressure has been found, we can add the pressure gradient to make the velocity field incompressible:

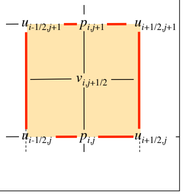
$$u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^{*} - \frac{\Delta t}{\frac{1}{2}(\rho_{i+1,j}^{n+1}+\rho_{i,j}^{n+1})}\frac{p_{i+1,j}-p_{i,j}}{\Delta x}$$

```
for i=2:nx, for j=2:ny+1   % CORRECT THE u-velocity
     u(i,j)=ut(i,j)-(dt*2.0/(r(i+1,j)+r(i,j)))*(p(i+1,j)-p(i,j))/dx;
end, end
```



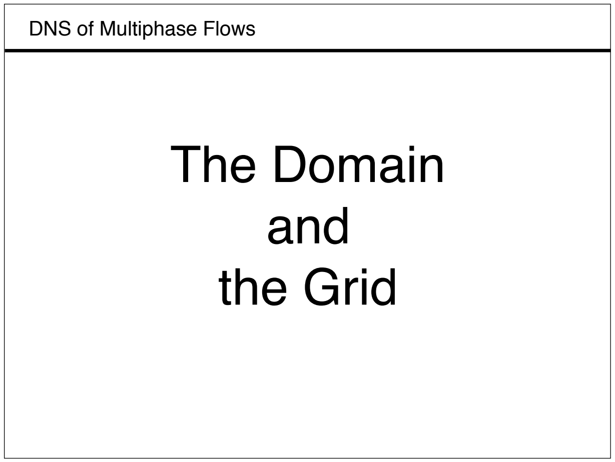
$$v_{i,j+1/2}^{n+1} = v_{i,j+1/2}^{*} - \frac{\Delta t}{\frac{1}{2}(\rho_{i,j+1}^{n+1}+\rho_{i,j}^{n+1})}\frac{p_{i,j+1}-p_{i,j}}{\Delta y}$$

```
for i=2:nx+1, for j=2:ny   % CORRECT THE v-velocity
     v(i,j)=vt(i,j)-(dt*2.0/(r(i,j+1)+r(i,j)))*(p(i,j+1)-p(i,j))/dy;
end, end
```



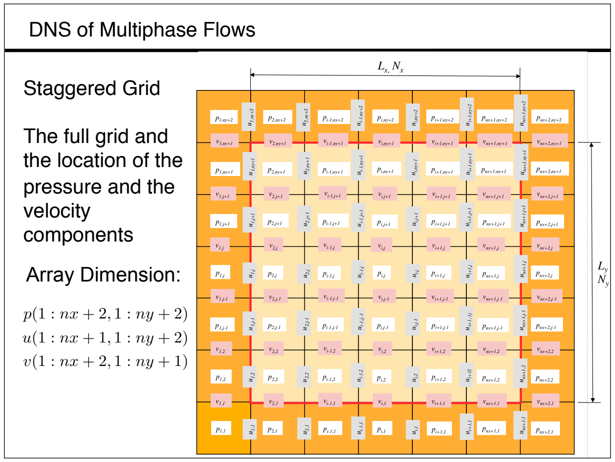
This completes the updating of the velocity field for the interior points, for the first order time integration that we are using here

14. The discrete equations for the corrected velocities are the same that we found earlier and used in the derivation of the pressure equation. The velocity at the new time step is the predicted velocity component minus delta t divided by the density at time n+1, interpolated to the velocity point, times a discrete approximation to the pressure, found by using the pressure at the boundaries of the velocity control volumes. Below the equation for each component we include code snippets for the update. As we did when finding the predicted velocities we do the update in separate loops, since the range of grid points is slightly different for each components, as we explain next.



The Domain and the Grid

15. To resolve the flow in a given domain we must divide the domain into the control volumes introduced above. In general, this can be a time-consuming and complex task, if the shape of the domain is complex.



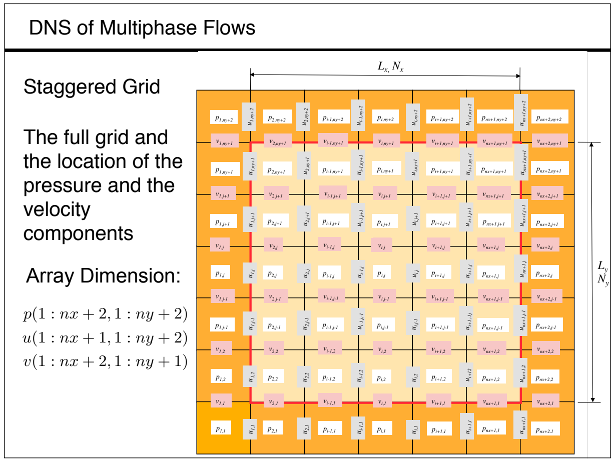
Staggered Grid

The full grid and the location of the pressure and the velocity components

Array Dimension:

$p(1:nx+2,1:ny+2)$
$u(1:nx+1,1:ny+2)$
$v(1:nx+2,1:ny+1)$

16-1. Here, however, we will examine the flow in a simple rectangular cavity, bounded by four walls, with at least one of the walls moving, to generate a flow. The size of the domain is Lx in the horizontal direction and Ly in the vertical one. The boundaries of the domain are shown by the read rectangle. We start by putting pressure control volumes or cells into the domain, starting at the lower left corner. We will assume that we need Nx volumes in the horizontal direction and Ny in the vertical direction. To simplify the implementation of our boundary conditions we will also put one row of pressure control volumes outside the domain. Thus, we will use Nx+2 control volumes in the horizontal direction and Ny+2 control volumes in the vertical direction. The control volumes outside the domain are generally referred to as ghost cells or ghost control volumes. Once we have decided where the pressure control volumes go, the velocity control volumes are set by shifting the pressure volumes half delta x to the right for the u velocity and half delta y up for the v-velocity.



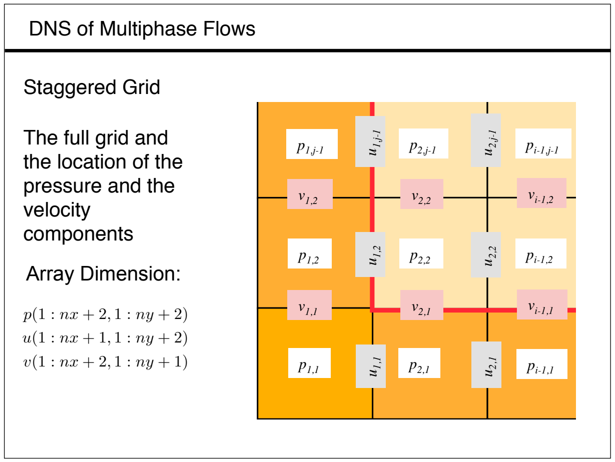
Staggered Grid

The full grid and the location of the pressure and the velocity components

Array Dimension:

$p(1:nx+2,1:ny+2)$
$u(1:nx+1,1:ny+2)$
$v(1:nx+2,1:ny+1)$

16-2. Notice that the first cell for the u velocity is now centered on the left boundary and that the ghost cell for the pressure for the left boundary is shifted to the right and is not needed. In the vertical direction the control volume for the u-velocity are not shifted with respect to the pressure cells. Thus, we use Nx+1 times Ny+2 cells for the u-velocity. Similarly, the first cell for the v-velocity is centered on the bottom boundary and the last shifted pressure cell is not needed. Therefore, we use Nx+2 times Ny+1 cells for the v-velocity.



DNS of Multiphase Flows

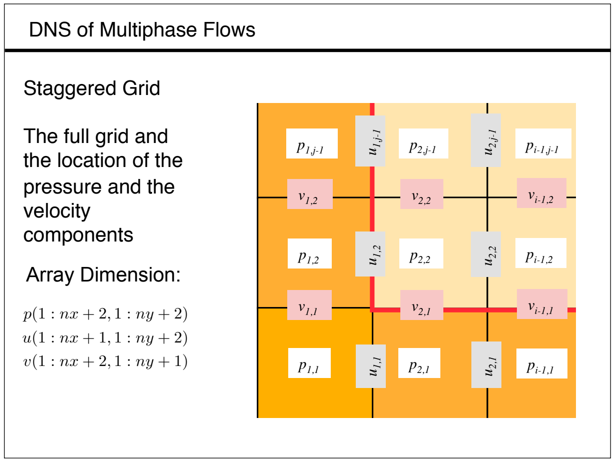
Staggered Grid

The full grid and the location of the pressure and the velocity components

Array Dimension:

$p(1:nx+2,1:ny+2)$
$u(1:nx+1,1:ny+2)$
$v(1:nx+2,1:ny+1)$

17-1. We start numbering the control volumes in the lower left hand corner, identifying the first pressure control volume, which is a ghost control volume outside the domain by 1,1. Since this cell does not border any interior velocity cell it is generally not needed. The ghost control volumes at the bottom are 1,2, 1,3 and on and those on the left are 2,1, 3,1 and so on. The first pressure control volume inside the domain is 2,2. For the u-velocity the control volumes have been shifted to the right so the first column on the left contains the normal velocities at the left boundary and are assumed to be given. The first row on the bottom are the ghost cells that are used to enforce the given tangent velocity at the bottom boundary. The first cell, 1,1 is located below the vertical left boundary and, since the velocities on the boundary are given, is usually not used. Similarly, the v-velocity control volumes have been shifted up so the first row on the bottom contains the normal velocities at the bottom boundary that are assumed to be given.



DNS of Multiphase Flows

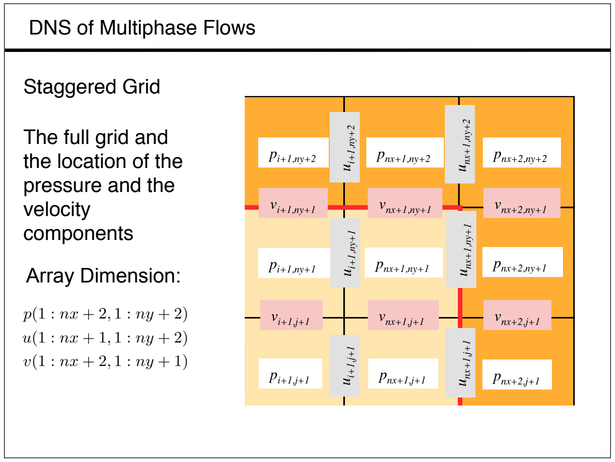
Staggered Grid

The full grid and the location of the pressure and the velocity components

Array Dimension:

$p(1:nx+2,1:ny+2)$
$u(1:nx+1,1:ny+2)$
$v(1:nx+2,1:ny+1)$

17-2. The first column on the left contains the ghost cells that are used to enforce the given tangent velocity at the left boundary. The first cell, 1,1 is located to the left of the bottom horizontal boundary and, since the velocities on the boundary are given, is usually not used.



DNS of Multiphase Flows

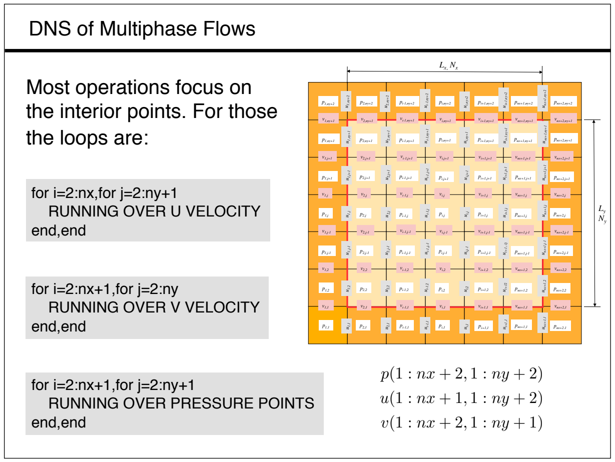
Staggered Grid

The full grid and the location of the pressure and the velocity components

Array Dimension:

$p(1:nx+2,1:ny+2)$
$u(1:nx+1,1:ny+2)$
$v(1:nx+2,1:ny+1)$

18. In the upper right side corner the last cell is the pressure ghost cell number Nx+2, NY+2, and since it does not border any interior velocity cell it is generally not needed. The pressure cells outside the boundary on the top and the bottom are the ghost cells for the pressure. The last row of u-velocities, on the top with i=Nx+1, are on the boundary and are given and the u-velocity in the ghost cells on the right, with j=Ny+2, are used to enforce the tangent velocities. Similarly, the last row of v-velocities, with j=Ny+1, are on the top boundary and are given and the v-velocity in the ghost cells on the right, with i=Nx+2 are used to enforce the tangent velocities.



DNS of Multiphase Flows

Most operations focus on the interior points. For those the loops are:

```
for i=2:nx,for j=2:ny+1
   RUNNING OVER U VELOCITY
end,end
```

```
for i=2:nx+1,for j=2:ny
   RUNNING OVER V VELOCITY
end,end
```

```
for i=2:nx+1,for j=2:ny+1
   RUNNING OVER PRESSURE POINTS
end,end
```

$$p(1:nx+2,1:ny+2)$$
$$u(1:nx+1,1:ny+2)$$
$$v(1:nx+2,1:ny+1)$$

19. The variables to be updated are generally those inside the domain. Since the numbers of pressure and velocity control volumes are different, we need to be careful to ensure that the indices take on the correct values. The quantities on the boundary, or in the ghost points, are either given, or treated in a special way. For the horizontal velocity we have i running over 2 to Nx and j running over 2 to Ny+1 and for the vertical velocities i runs over 2 to Nx+1 and j over 2 to Ny. For the pressure, i takes on the values 2 to Nx+1 and j takes on 2 to Ny+1.