Solutions Manual for

Fluid Mechanics: Fundamentals and Applications

Third Edition

Yunus A. Çengel & John M. Cimbala

McGraw-Hill, 2013

Chapter 15
INTRODUCTION TO COMPUTATIONAL FLUID DYNAMICS

Fundamentals, Grid Generation, and Boundary Conditions

15-1C

Solution We are to list the unknowns and the equations for a given flow situation.

Analysis There are only three unknowns in this problem, ***u, v,* and *P* (or*P′*)**. Thus, we require three equations: **continuity**, ***x* momentum** (or *x* component of Navier-Stokes), and ***y* momentum** (or *y* component of Navier-Stokes). These equations, when combined with the appropriate boundary conditions, are sufficient to solve the problem.

Discussion The actual equations to be solved by the computer are discretized versions of the differential equations.

15-2C

Solution We are to define several terms or phrases and provide examples.

Analysis

(*a*) **A computational domain is a region in space (either 2-D or 3-D) in which the numerical equations of fluid flow are solved by CFD**. The computational domain is bounded by edges (2-D) or faces (3-D) on which boundary conditions are applied.

(*b*) **A mesh is generated by dividing the computational domain into tiny cells**. The numerical equations are then solved in each cell of the mesh. A mesh is also called a grid.

(*c*) **A transport equation is a differential equation representing how some property is transported through a flow field**. The transport equations of fluid mechanics are conservation equations. For example, the continuity equation is a differential equation representing the transport of mass, and also conservation of mass. The Navier-Stokes equation is a differential equation representing the transport of linear momentum, and also conservation of linear momentum.

(*d*) **Equations are said to be coupled when at least one of the variables (unknowns) appears in more than one equation**. In other words, the equations cannot be solved alone, but must be solved simultaneously with each other. This is the case with fluid mechanics since each component of velocity, for example, appears in the continuity equation and in all three components of the Navier-Stokes equation.

Discussion Students' definitions should be in their own words.

15-3C

Solution We are to discuss the difference between nodes and intervals and analyze a given computational domain in terms of nodes and intervals.

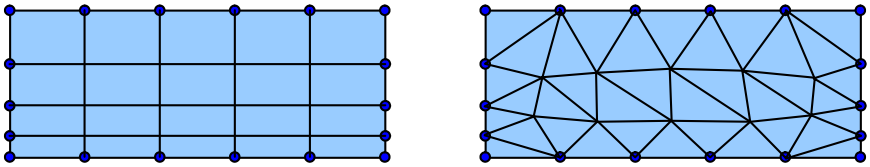
Analysis **Nodes are points along an edge of a computational domain that represent the vertices of cells**. In other words, they are the points where corners of the cells meet. **Intervals, on the other hand, are short line segments between nodes**. Intervals represent the small edges of cells themselves. In Fig. P15-3 there are **6 nodes** and **5 intervals** on the top and bottom edges. There are **5 nodes** and **4 intervals** on the left and right edges.

Discussion We can extend the node and interval concept to three dimensions.

15-4C

Solution For a given computational domain with specified nodes and intervals we are to compare a structured grid and an unstructured grid and discuss.

Analysis We construct the two grids in the figure: (a) structured, and (b) unstructured.



(a) (b)

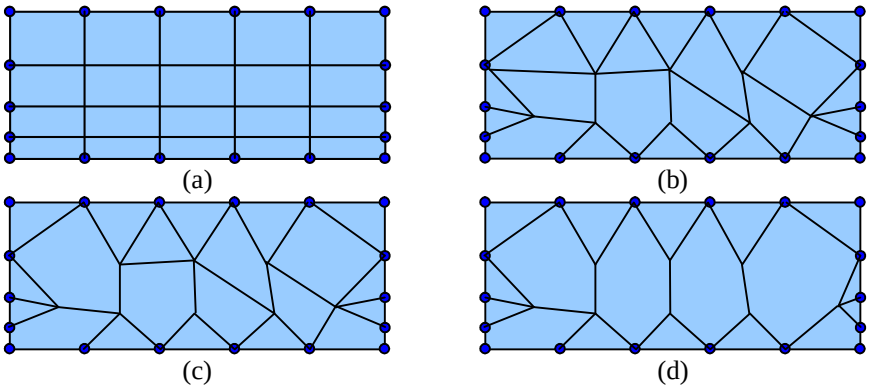
There are $5 \times 4 = 20$ cells in the structured grid, and there are 36 cells in the unstructured grid.

Discussion Depending on how individual students construct their unstructured grid, the shape, size, and number of cells may differ considerably.

15-5C

Solution For a given computational domain with specified nodes and intervals we are to compare a structured mesh and a polyhedral mesh and discuss.

Analysis We construct the two grids in the figure: (a) structured, and (b) unstructured polyhedral. We show two other options in (c) and (d). There are many possible answers for the polyhedral mesh, depending on how large you want your cells to be.



(a) (b)

(c) (d)

There are $5 \times 4 = 20$ cells in the structured grid. There are 22 cells in polyhedral grid (b). There are some cells with 3 sides, 4 sides, and 5 sides, as required. Compared to the triangular mesh with 36 cells, we have reduced the cell count considerably. In (c) and (d), there are 21 cells and 18 cells respectively. In case (d) we have reduced the cell count below that of even the structured grid. In that case, 3 of the cells have 6 sides each. The cell reduction is particularly useful in large 3-D problems where CPU time and computer memory are important limitations.

Discussion Note that the node distribution along the boundaries is identical in each case, but we have great flexibility in how we create the grid. Depending on how individual students construct their unstructured grid, the shape, size, and number of cells may differ considerably.

15-6C

Solution We are to summarize the eight steps involved in a typical CFD analysis.

Analysis We list the steps in the order presented in this chapter:

1. **Specify a computational domain and generate a grid.**
2. **Specify boundary conditions on all edges or faces.**
3. **Specify the type of fluid and its properties.**
4. **Specify numerical parameters and solution algorithms.**
5. **Apply initial conditions as a starting point for the iteration.**
6. **Iterate towards a solution.**
7. **After convergence, analyze the results (post processing).**
8. **Calculate global and integral properties as needed.**

Discussion The order of some of the steps is interchangeable, particularly Steps 2 through 5.

15-7C

Solution We are to explain why the cylinder should not be centered horizontally in the computational domain.

Analysis Flow separates over bluff bodies, generating a wake with **reverse flow and eddies downstream of the body**. There are no such problems upstream. Hence it is always wise to extend the downstream portion of the domain as far as necessary to avoid reverse flow problems at the outlet boundary.

Discussion The same problems arise at the outlet of ducts and pipes – sometimes we need to extend the duct to avoid reverse flow at the outlet boundary.

15-8C

Solution We are to discuss the significance of several items with respect to iteration.

Analysis

(*a*) In a CFD solution, we typically iterate towards a solution. **In order to get started, we make some initial conditions for all the variables (unknowns) in the problem**. These initial conditions are wrong, of course, but they are necessary as a starting point. Then we begin the iteration process, eventually obtaining the solution.

(*b*) **A residual is a measure of how much our variables differ from the "exact" solution**. We construct a residual by putting all the terms of a transport equation on one side, so that the terms all add to zero if the solution is correct. As we iterate, the terms will *not* add up to zero, and the remainder is called the residual. As the CFD solution iterates further, the residual should (hopefully) decrease.

(*c*) **Iteration is the numerical process of marching towards a final solution**, beginning with initial conditions, and progressively correcting the solution. As the iteration proceeds, the variables converge to their final solution as the residuals decrease.

(*d*) **Once the CFD solution has converged, post processing is performed on the solution**. Examples include plotting velocity and pressure fields, calculating global properties, generating other flow quantities like vorticity, etc. Post processing is performed after the CFD solution has been found, and does not change the results. Post processing is generally not as CPU intensive as the iterative process itself.

Discussion We have assumed steady flow in the above discussions.

15-9C

Solution We are to discuss how the iteration process is made faster.

Analysis

(*a*) With *multigridding*, **solutions of the equations of motion are obtained on a coarse grid first, followed by successively finer grids**. This speeds up convergence because the gross features of the flow are quickly established on the coarse grid, and then the iteration process on the finer grid requires less time.

(*b*) In some CFD codes, **a steady flow is treated as though it were an unsteady flow. Then, an *artificial time* is used to march the solution in time**. Since the solution is steady, however, the solution approaches the steady-state solution as "time" marches on. In some cases, this technique yields faster convergence.

Discussion There are other "tricks" to speed up the iteration process, but CFD solutions often take a long time to converge.

15-10C

Solution We are to list the boundary conditions that are applicable to a given edge, and we are to explain why other boundary conditions are not applicable.

Analysis We may apply the following boundary conditions: **outflow**, **pressure inlet**, **pressure outlet**, **symmetry** (to be discussed), **velocity inlet**, and **wall**. The curved edge cannot be an *axis* because an axis must be a straight line. The edge cannot be a *fan* or *interior* because such edges cannot be at the outer boundary of a computational domain. Finally, **the edge cannot be periodic since there is no other edge along the boundary of the computational domain that is of identical shape** (a periodic boundary must have a "partner"). The symmetry boundary condition merits further discussion. Numerically, gradients of flow variables in the direction normal to a symmetry boundary condition are set to zero, and there is no mathematical reason why the curved right edge of the present computational domain cannot be set as symmetry. However, you would be hard pressed to think of a physical situation in which a curved edge like that of Fig. P15-9 would be a valid symmetry boundary condition.

Discussion Just because you can set a boundary condition and generate a CFD result does not guarantee that the result is physically meaningful.

15-11C

Solution We are to discuss the standard method to test for adequate grid resolution.

Analysis The standard method to test for adequate grid resolution is to **increase the resolution** (by a factor of 2 in all directions if feasible) **and repeat the simulation**. If the results do not change appreciably, the original grid is deemed adequate. If, on the other hand, there are significant differences between the two solutions, the original grid is likely of inadequate resolution. In such a case, an even finer grid should be tried until the grid is adequately resolved.

Discussion Keep in mind that if the boundary conditions are not specified properly, or if the chosen turbulence model is not appropriate for the flow being simulated by CFD, no amount of grid refinement is going to make the solution more physically correct.

15-12C

Solution We are to discuss the difference between a pressure inlet boundary condition and a velocity inlet boundary condition, and we are to explain why both pressure and velocity cannot be specified on the same boundary.

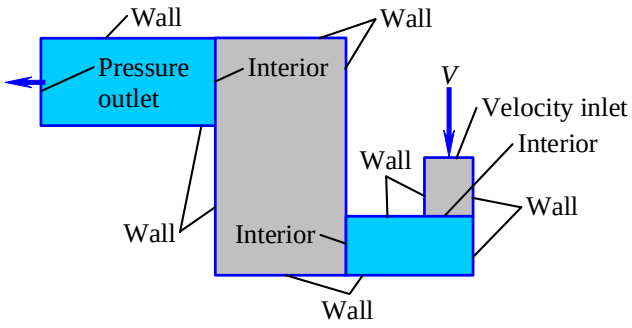
Analysis **At a pressure inlet we specify the pressure but not the velocity**. **At a velocity inlet we specify the opposite – velocity but not pressure**. To specify both pressure and velocity would lead to mathematical **over-specification, since pressure and velocity are coupled in the equations of motion**. When pressure is specified at a pressure inlet (or outlet), the CFD code automatically adjusts the velocity at that boundary. In a similar manner, when velocity is specified at a velocity inlet, the CFD code adjusts the pressure at that boundary.

Discussion Since pressure and velocity are coupled, specification of both at a boundary would lead to inconsistencies in the equations of motion at that boundary.

15-13C

Solution We are to label all the boundary conditions to be applied to a computational domain.

Analysis The inlet is a *velocity inlet*. The outlet is a *pressure outlet*. All other edges that define the outer limits of the computational domain are *walls*. Finally, there are three edges that must be specified as *interior*. These are all labeled in the figure below.



Discussion It is critical that each boundary condition be specified carefully. Otherwise the CFD solution will not be correct.

15-14C

Solution We are to analyze what will happen to inlet pressure and outlet velocity when a fan is turned on in the computational domain of the previous problem.

Analysis Since the fan helps to push air through the channel, the inlet pressure will adjust itself so that less inlet pressure is required. In other words, the **inlet pressure will decrease when the fan is turned on**. Since the inlet velocity is the same in both cases, the mass flow rate (and volume flow rate since the flow is incompressible) must remain the same for either case. Therefore, **outlet velocity will not change**.

Discussion It may seem at first glance that V_{out} should increase because of the fan, but in order to conserve mass, the outlet velocity cannot change. The solution is constrained by the specified inlet boundary condition. In a real physical experiment, there is no such restriction. The fan would cause the inlet pressure to decrease, the inlet velocity to increase, and the outlet velocity to increase.

15-15C

Solution We are to list and briefly describe six boundary conditions, and we are to give an example of each.

Analysis In the chapter we list ten, so any six of these will suffice:

- **Axis**: Used in axisymmetric flows as the axis of rotation. Example: the axis of a torpedo-shaped body.
- **Fan**: An internal edge (2-D) or face (3-D) across which a sudden pressure rise is specified. Example: an axial flow fan in a duct.
- **Interior**: An internal edge (2-D) or face (3-D) across which nothing special happens – the interior boundary condition is used at the interface between two blocks. Example: all of the multiblock problems in this chapter, which require this boundary condition at the interface between any two blocks.
- **Outflow**: An outlet boundary condition in which the gradient of fluid properties is zero normal to the outflow boundary. Outflow is typically useful far away from the object or area of interest in a flow field. Example: the far field of flow over a body.
- **Periodic**: When the physical geometry has periodicity, the periodic boundary condition is used to specify that whatever passes through one face of the periodic pair must simultaneously enter the other face of the periodic pair. Example: in a heat exchanger where there are several rows of tubes.
- **Pressure inlet**: An inflow boundary in which pressure (but not velocity) is known and specified across the face. Example: the high pressure settling chamber of a blow-down wind tunnel facility.
- **Pressure outlet**: An outflow boundary in which pressure (but not velocity) is known and specified across the face. Example: the outlet of a pipe exposed to atmospheric pressure.
- **Symmetry**: A face over which the gradients of all flow variables are set to zero normal to the face – the result is a mirror image across the symmetry plane. Fluid cannot flow *through* a symmetry plane. Example: the mid-plane of flow over a circular cylinder in which the lower half is a mirror image of the upper half.
- **Velocity inlet**: An inflow boundary condition in which velocity (but not pressure) is known and specified across the face. Example: a uniform freestream inlet flow entering a computational domain from one side.
- **Wall**: A boundary through which fluid cannot pass and at which the no-slip condition (or a shear stress condition) is applied. Example: the surface of an airfoil that is being modeled by CFD.

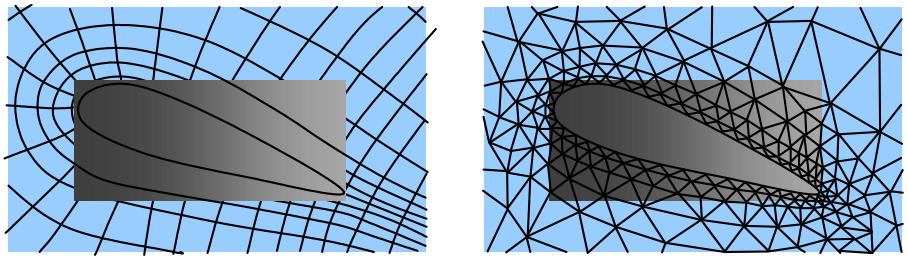
Discussion There are additional boundary conditions used in CFD calculations, but these are the only ones discussed in this chapter.

15-16

Solution We are to sketch a structured and an unstructured grid near the airfoil surface, and discuss advantages and disadvantages of each.

Analysis In either case it is wise to cluster cells close to the airfoil surface since we expect that a thin boundary layer will exist along the surface, and we need many tiny cells within the boundary layer to adequately resolve it. Some simple, coarse meshes are drawn in Fig. 1. We would certainly want much higher resolution for CFD calculations.

FIGURE 1
A coarse structured (a) and unstructured (b) grid. Notice that the cells are clustered (more fine) near the surface of the airfoil since there is likely to be large velocity gradients there (in the boundary layer).



(a) (b)

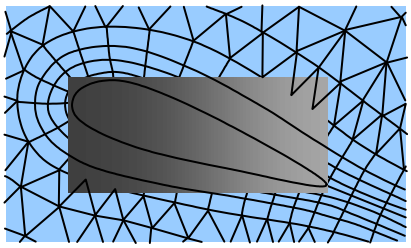
The structured grid in Fig 1a is called a C-grid since it wraps around the airfoil like the letter "C". The main advantage of the structured grid is that we can get high resolution near the surface with few cells. The main advantage of the unstructured grid is that it is somewhat easier to generate when the geometry is complicated (especially for highly curved surfaces). Furthermore, it is easier to transition between curved and straight edges with an unstructured grid. The main disadvantage of an unstructured grid is that more cells are required for the same spatial resolution.

Discussion There are numerous other ways to construct a grid around this airfoil.

15-17

Solution We are to sketch a hybrid grid around an airfoil and explain its advantages.

Analysis We sketch a hybrid grid in the figure. Note that the grid is structured near the airfoil surface, but unstructured beyond the surface. **The advantage of a hybrid grid is that it combines the advantages of both structured and unstructured grids**. Near surfaces we can use a structured grid to finely resolve the boundary layer with a minimum number of cells, and away from surfaces we can use an unstructured grid so that we can rapidly expand the cell size. We can also more easily blend the grid into the edges of the computational domain with an unstructured grid.



Discussion A structured grid is generally the best choice, but a hybrid grid is often a better option than a fully unstructured grid.

15-18

Solution We are to sketch the blocking for a structured grid, sketch a coarse grid, and label all the boundary conditions to be applied to the computational domain.

Analysis First of all, we recognize that because of symmetry, we can split the domain in half vertically. We construct four blocks around the half-cylinder to transform from round to square, and then we add simple rectangular blocks upstream and downstream of the cylinder (Fig. 1). There is a total of **six blocks**.

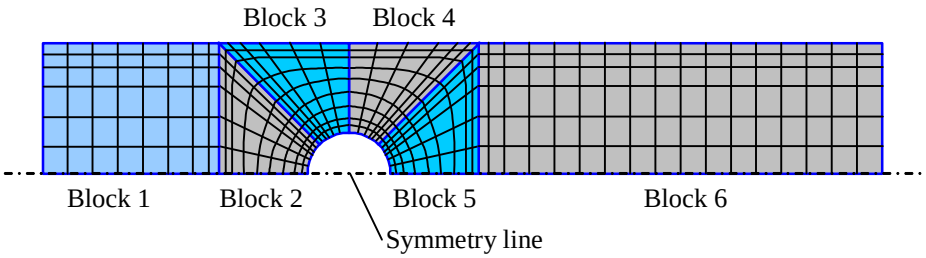


FIGURE 1
A possible blocking topology and coarse structured grid for a 2-D multiblock computational domain.

With the block structure of Fig. 1 no cells are highly skewed, and cells are clustered near the cylinder wall and the upper wall of the duct as desired.

The bottom edge of the computational domain is a line of **symmetry**. The inlet is a **velocity inlet**. The outlet is a **pressure outlet**. The upper edge of the computational domain is a **wall**. The edges that define the cylinder are also **walls**. Finally, there are 5 edges that are specified as **interior**. These are all labeled in Fig. 2.

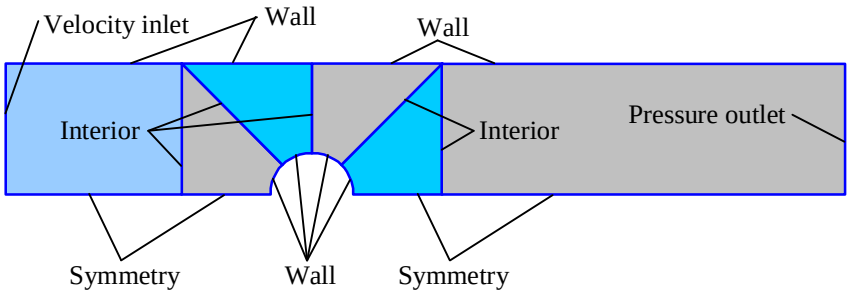


FIGURE 2
Boundary conditions specified on each edge of a 2-D multiblock computational domain.

Discussion There are alternative ways to set up the blocking topology. For example, at the top we may define a thin block (Block 7) that stretches across the entire horizontal domain so that the boundary layer on the top wall of the channel can be more adequately resolved (Fig. 3).

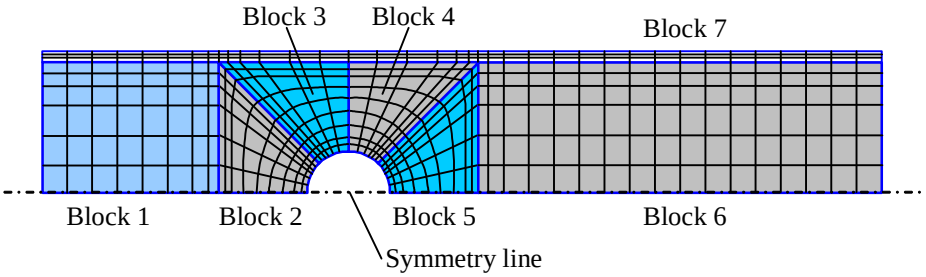


FIGURE 3
An alternative blocking topology and coarse structured grid for the 2-D multiblock computational domain. A seventh block is added at the top for better boundary layer resolution near the top plate.

15-19

Solution We are to sketch the blocking for a structured grid, sketch a coarse grid, and label all the boundary conditions to be applied to the computational domain.

Analysis First of all, we recognize that because of symmetry, we can split the domain in half vertically. We construct four blocks inside the half-circle, and then we add blocks with one curved edge and three straight edges upstream and downstream of the cylinder (Fig. 1).

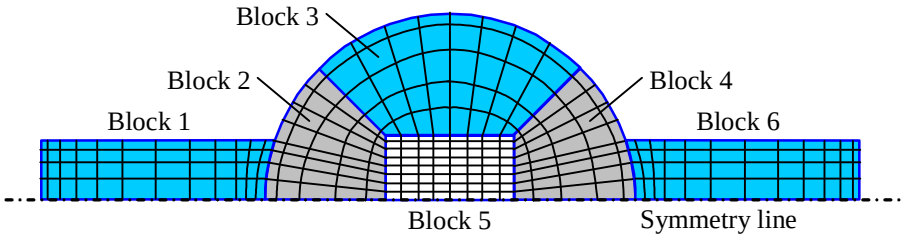


FIGURE 1
The blocking and coarse structured grid for a 2-D multiblock computational domain.

The setup of Fig. 1 contains six blocks. With this block structure, no cells are highly skewed, and cells are clustered near upper wall of the duct as desired. Cells are also clustered at the junctions between Blocks 1 and 2 and Blocks 4 and 6, where flow separation may occur.

The bottom edge of the computational domain is a line of **symmetry**. The inlet is a **velocity inlet**. The outlet is a **pressure outlet**. The upper edge of the computational domain is a **wall**. The edges that define the cylinder are also **walls**. Finally, there are 5 edges that are specified as **interior**. These are all labeled in Fig. 2.

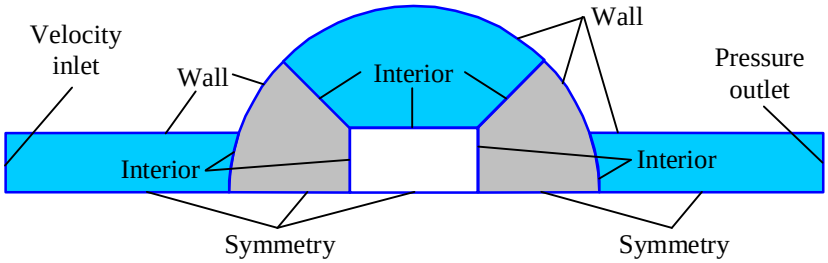


FIGURE 2
Boundary conditions specified on each edge of a 2-D multiblock computational domain.

Discussion There are of course, alternative ways to set up the blocking topology.

15-20

Solution We are to modify an existing grid so that all blocks are elementary blocks. Then we are to verify that the total number of cells does not change.

Analysis The right edge of Block 2 of Fig. 15-11b is split twice to accommodate Block 1. We therefore split Block 2 into three separate elementary blocks. Unfortunately, this process ends up splitting Block 6 twice, which in turn splits Block 4 twice. **We end up with 12 elementary blocks** as shown in Fig. 1.

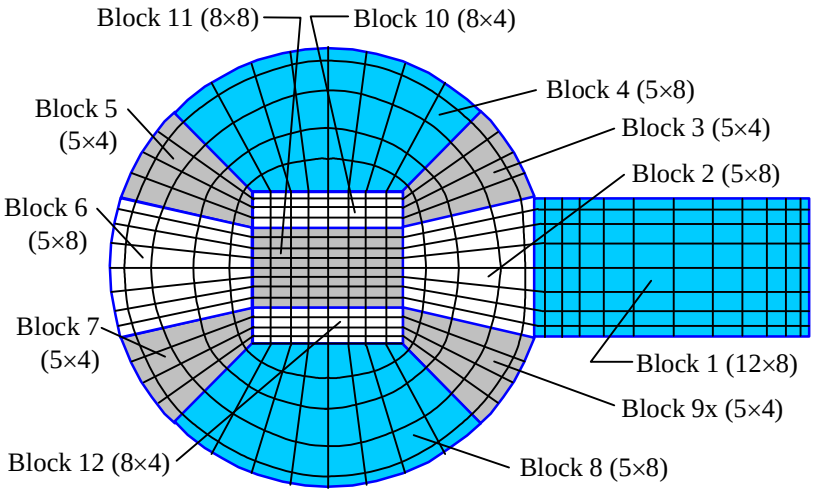


FIGURE 1
The blocking and coarse structured grid for a 2-D multiblock computational domain. Only elementary blocks are used in this grid.

We add up all the cells in these 12 blocks – we get a total of **464 cells**. This agrees with the total of 464 cells for the original 6 blocks in the domain.

Discussion Sometimes it is easier to create a grid with elementary blocks, even if the CFD code can accept blocks with split edges or faces.

15-21

Solution We are to modify an existing grid into a smaller number of non- elementary blocks, and we are to verify that the total number of cells does not change.

Analysis We combine Blocks 2, 3, 4, and 5 of Fig. 15-10b. Together, these produce one structured grid that wraps around the square in the middle – there are still 5 *i* intervals, but now there are 48 *j* intervals. **We end up with 3 non-elementary blocks**, as shown in Fig. 1.

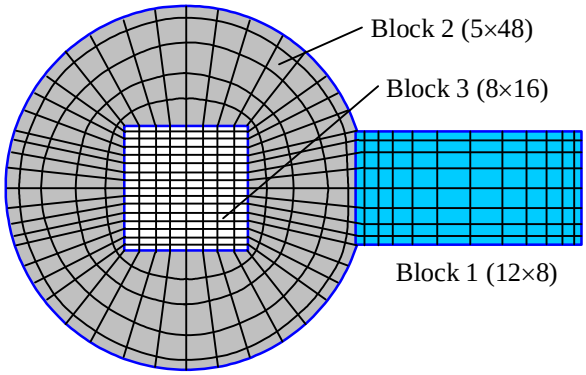


FIGURE 1
The blocking and coarse structured grid for a 2-D multiblock computational domain. Only elementary blocks are used in this grid.

We add up all the cells in these 12 blocks – we get a total of **464 cells**. This agrees with the total of 464 cells for the original 6 blocks in the domain.

Discussion Block 2 in Fig. 1 is called an **O-grid** (for obvious reasons).

15-22

Solution We are to generate a computational domain and label all appropriate boundary conditions for one stage of a new heat exchanger design.

Analysis We take advantage of the periodicity of the geometry. There are several ways to create a periodic grid for this flow. The simplest computational domain consists of a single flow passage between two neighboring tubes. We can make the periodic edge intercept anywhere on the front portion of the tube that we desire. We choose the lower surface for convenience and simplicity. The periodic computational domain is sketched in Fig. 1.

Boundary conditions are also straightforward, and are labeled in Fig. 1. For a known inlet velocity we set the boundary condition at the left edge as a **velocity inlet**. The tube walls are obviously set as **walls**. The outlet can be set as either a **pressure outlet** or an **outflow**, depending on the provided information and how far the outlet region extends beyond the tubes. Finally, we set two pairs of **translationally periodic** boundaries, one fore and one aft of the tubes. We label them separately to avoid confusion.

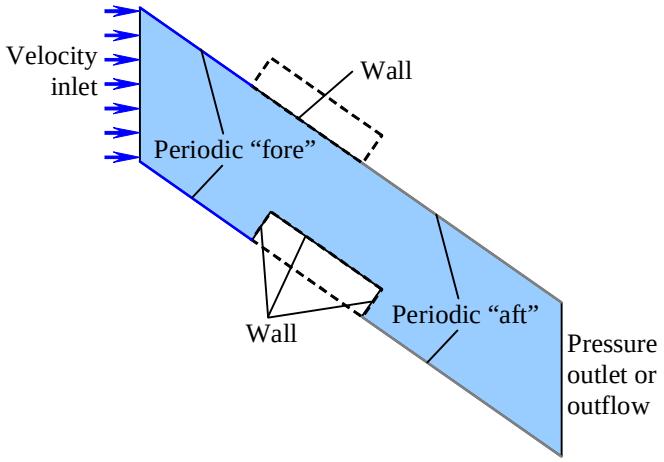


FIGURE 1
A periodic computational domain for a given geometry. Boundary conditions are also labeled.

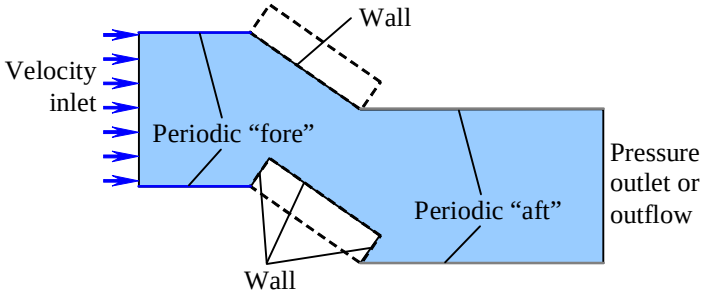


FIGURE 2
An alternative periodic computational domain for a given geometry. Boundary conditions are also labeled.

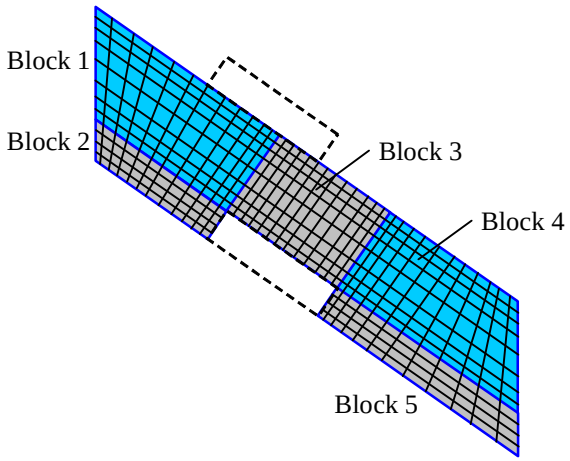
Discussion The fore and aft periodic edges are not horizontal in Fig. 1. This is not a problem since the periodic boundary condition is not restricted to horizontal or even to flat surfaces. An alternative, equally acceptable computational domain is shown in Fig. 2.

15-23

Solution We are to sketch a structured multiblock grid with four-sided elementary blocks for a given computational domain.

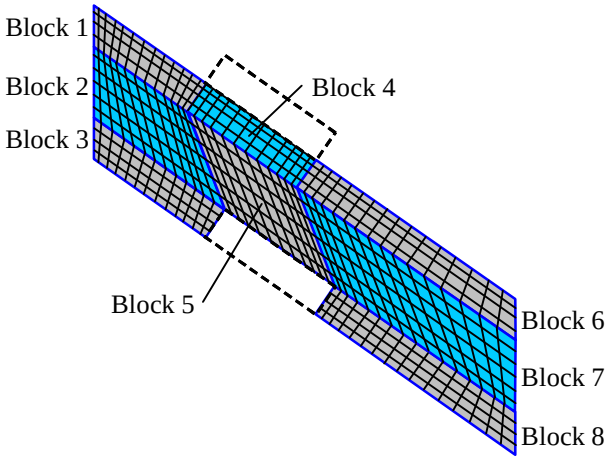
Analysis We choose the computational domain of Fig. 1 of the previous problem. Since all edges are straight, the blocking scheme can be rather simple. We sketch the blocking topology and apply a coarse mesh in Fig. 1 for the case in which the CFD code does not require the node distribution to be exactly the same on periodic pairs.

FIGURE 1
The blocking topology and a coarse structured grid for a periodic computational domain. This blocking topology applies to CFD codes that allow a block's edges to be split for application of boundary conditions, and do *not* require periodic edge pairs to have identical node distributions.



Unfortunately, many CFD codes require that the node distribution on periodic pairs of edges be identical (the two edges of a periodic pair are "linked" in the grid generation process). In such a case, the grid of Fig. 1 would not be acceptable. Furthermore, although the edges of the blocks of Fig. 1 are not split with respect to adjacent blocks, the top edges of Block 1 and Block 3 are split with respect to the boundary conditions (part of the edge is periodic and part is a wall). Thus these blocks are not really elementary blocks after all. We construct a more elaborate blocking topology in Fig. 2 to correct these problems. The node distribution on the edges of each periodic pair are identical, at the expense of more complexity (7 instead of 5 blocks) and more cell skewness.

FIGURE 2
The blocking topology and a coarse structured grid for a periodic computational domain. This blocking topology applies to CFD codes that require elementary blocks and require periodic edge pairs to have identical node distributions.



Discussion Some of the cells have moderate skewness with the blocking topology of Fig. 2, especially near the corners of Block 2 and Block 7 and throughout Block 5. A more complicated topology can be devised to reduce the amount of skewness.

15-24

Solution We are to discuss why there is reverse flow in this CFD calculation, and then we are to explain what can be done to correct the problem.

Analysis **Reverse flow at an outlet is usually an indication that the computational domain is not large enough**. In this case the rectangular heat exchanger tubes are inclined at 35°, and the flow will most certainly separate, leaving large recirculating eddies in the wakes. **Anita should extend the computational domain in the horizontal direction downstream** so that the eddies have a chance to "close" and the flow has a chance to re-develop into a flow without any reverse flow.

Discussion In most commercial CFD codes a warning will pop up on the computer monitor whenever there is reverse flow at an outlet. This is usually an indication that the computational domain should be enlarged.

15-25

Solution We are to generate a computational domain and label all appropriate boundary conditions for two stages of a heat exchanger.

Analysis We look for the smallest computational domain that takes advantage of the periodicity of the geometry. There are several ways to create a periodic grid for this flow. The simplest computational domain consists of a single flow passage between two neighboring tubes. We can make the periodic edge intercept anywhere on the fore and aft portions of the heat exchanger that we desire. We choose the periodic computational domain sketched in Fig. 1.

Boundary conditions are also straightforward, and are labeled in Fig. 1. For a known inlet velocity we set the boundary condition at the left edge as a **velocity inlet**. The tube walls are obviously set as **walls**. The outlet can be set as either a **pressure outlet** or an **outflow**, depending on the provided information and how far the outlet region extends beyond the tubes. Finally, we set three pairs of **translationally periodic** boundaries, one fore, one mid, and one aft of the tubes. We label them separately to avoid confusion.

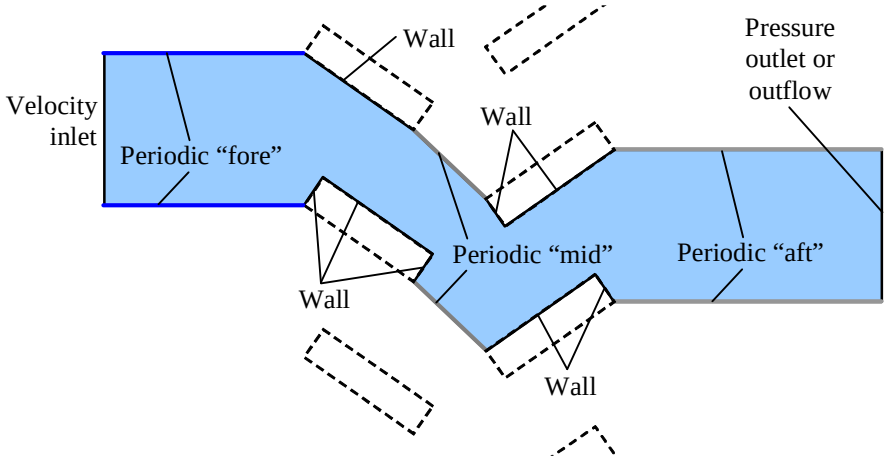


FIGURE 1
A periodic computational domain for a given geometry. Boundary conditions are also labeled.

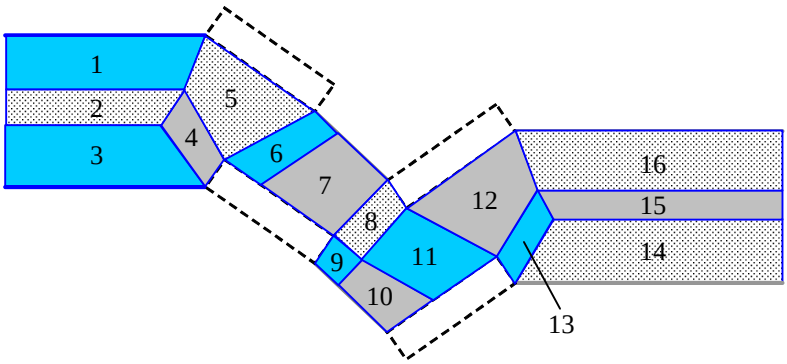
Discussion Many other equally acceptable computational domains are possible.

15-26

Solution We are to sketch a structured multiblock grid with four-sided elementary blocks for a given computational domain.

Analysis We choose the computational domain of Fig. 1 of the previous problem. We sketch one possible elementary blocking topology in Fig. 1 for the case in which the CFD code requires the node distribution to be exactly the same on periodic pairs. We also assume that we cannot split one periodic edge and not its partner. The blocks are numbered. This topology has **16 elementary blocks**.

FIGURE 1
The elementary blocking topology for a periodic computational domain. This blocking topology applies to CFD codes that do not allow a block's edges to be split for application of boundary conditions, and requires periodic edge pairs to have identical node distributions.



Note that with this blocking topology we had to split the periodic "mid" boundary pair into two edges (the tops of blocks 6 and 7 and the bottoms of blocks 9 and 10). As long as both pairs of each segment are the same size and have the same number of nodes, this is not a problem. In the CFD code we would have to name each periodic pair separately, however. The block numbers are labeled. Notice that most of the blocks are nearly rectangular such that none of the computational cells would have to be highly skewed.

Discussion This seems like a rather complicated blocking topology. It would require a bit of work to generate the grid. However, the time spent on developing a good grid is usually well worth the effort. By reducing the amount of cell skewness, we are able to speed up the CFD calculations and obtain more accurate results. This kind of topology also enables us to cluster cells near walls and wakes as needed.

General CFD Problems

15-27

Solution We are to generate three different coarse grids for the same geometry and node distribution, and then compare the cell count and grid quality.

Analysis The three meshes are shown in Fig. 1. The node distributions along the edges of the computational domain are identical in all three cases, and no smoothing of the mesh is performed. The structured multi-block mesh is shown in Fig. 1a. We split the domain into four blocks for convenience, and to achieve cells with minimal skewing. There are 1060 cells. The unstructured triangular mesh is shown in Fig. 1b. There is only one block, and it contains 1996 cells. The unstructured quad mesh is shown in Fig. 1c. It has 833 cells in its one block. Comparing the three meshes, the triangular unstructured mesh has too many cells. The unstructured quad mesh has the least number of cells, but the clustering of cells occurs in undesirable locations, such as at the outlets on the right. The structured quad mesh seems to be the best choice for this geometry – it has only about 27% more cells than the unstructured quad mesh, but we have much more control on the clustering of the cells. Skewness is not a problem with any of the meshes.

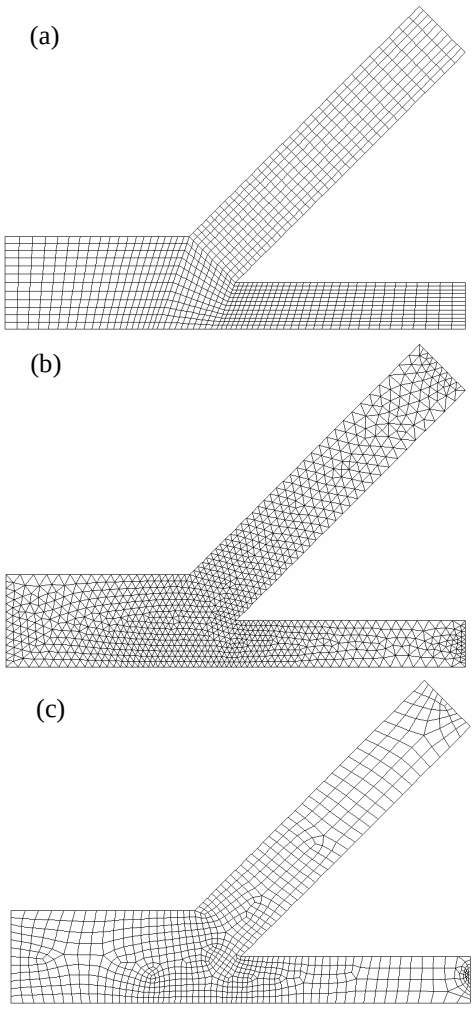


FIGURE 1
Comparison of three meshes: (a) structured multiblock, (b) unstructured triangular, and (c) unstructured quadrilateral.

Discussion Depending on the grid generation software and the specified node distribution, students will get a variety of results.

15-28

Solution We are to run a laminar CFD calculation of flow through a wye, calculate the pressure drop and how the flow splits between the two branches.

Analysis We choose the structured grid for our CFD calculations. The back pressure at both outlets is set to zero gage pressure, and the average pressure at the inlet is calculated to be -8.74×10^{-5} Pa. The pressure drop through the wye is thus only 8.74×10^{-5} Pa (a negligible pressure drop). The streamlines are shown in Fig. 1. For this case, 57.8% of the flow goes out the upper branch, and 42.2% goes out the lower branch.

Discussion There appears to be some tendency for the flow to separate at the upper left corner of the branch, but there is no reverse flow at the outlet of either branch. This case is compared to a turbulent flow case in the following problem.

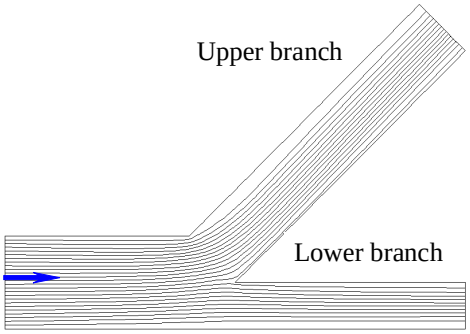


FIGURE 1
Streamlines for laminar flow through a wye.

15-29

Solution We are to run a turbulent CFD calculation of flow through a wye, calculate the pressure drop and how the flow splits between the two branches.

Analysis We choose the structured grid for our CFD calculations. The back pressure at both outlets is set to zero gage pressure, and the average pressure at the inlet is calculated to be -3.295 Pa. The pressure drop through the wye is thus 3.295 Pa (a significantly higher pressure drop than that of the laminar flow, although we note that the inlet velocity for the laminar flow case was 0.002 times that of the turbulent flow case). The streamlines are shown in Fig. 1. For this case, 54.4% of the flow goes out the upper branch, and 45.6% goes out the lower branch. Compared to the laminar case, a greater percentage of the flow goes out the lower branch for the turbulent case. The streamlines at first look similar, but a closer look reveals that the spacing between streamlines in the turbulent case is more uniform, indicating that the velocity distribution is also more uniform (more "full"), as is expected for turbulent flow.

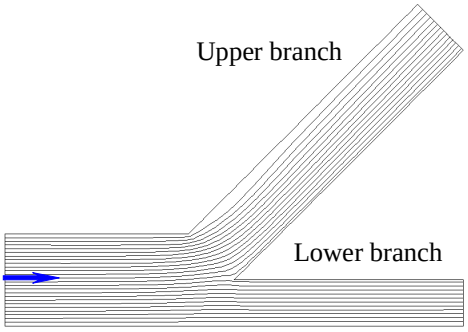


FIGURE 1
Streamlines for turbulent flow through a wye. The k-ε turbulence model is used.

Discussion There appears to be some tendency for the flow to separate at the upper left corner of the branch, but there is no reverse flow at the outlet of either branch.

15-30

Solution We are to keep refining a grid until it becomes grid independent for the case of a laminar boundary layer.

Analysis Students will have varied results, depending on the grid generation code, CFD code, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-31

Solution We are to keep refining a grid until it becomes grid independent for the case of a turbulent boundary layer.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-32

Solution We are to study ventilation in a simple 2-D room using CFD, and using a structured rectangular grid.

Analysis Students will have varied results, depending on the grid generation code, CFD code, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-33

Solution We are to repeat the previous problem except use an unstructured grid, and we are to compare results.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-34

Solution We are to use CFD to analyze the effect of moving the supply and/or return vents in a room.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-35

Solution We are to use CFD to analyze a simple 2-D room with air conditioning and heat transfer.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-36

Solution We are to compare the CFD predictions for 2-D and 3-D ventilation.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-37

Solution We are to use CFD to study compressible flow through a converging nozzle with inviscid walls. Specifically, we are to vary the pressure until we have choked flow conditions.

Analysis Students will have varied results, depending on the grid generation code, CFD code, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-38

Solution We are to repeat the previous problem, but allow friction at the wall, and also use a turbulence model. We are then to compare the results to those of the previous problem to see the effect of wall friction and turbulence on the flow.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-39

Solution We are to generate a low-drag, streamlined, 2-D body, and try to get the smallest drag in laminar flow.

Analysis Students will have varied results, depending on the grid generation code, CFD code, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-40

Solution We are to generate a low-drag, streamlined, axisymmetric body, and try to get the smallest drag in laminar flow. We are also to compare the axisymmetric case to the 2-D case of the previous problem.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-41

Solution We are to generate a low-drag, streamlined, axisymmetric body, and try to get the smallest drag in turbulent flow. We are also to compare the turbulent drag coefficient to the laminar drag coefficient of the previous problem.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-42

Solution We are to use CFD to study Mach waves in supersonic flow. We are also to compare the computed Mach angle with that predicted by theory.

Analysis Students will have varied results, depending on the grid generation code, CFD code, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-43

Solution We are to study the effect of Mach number on the Mach angle in supersonic flow, and we are to compare to theory.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

Review Problems

15-44C

Solution

(*a*) ***False***: If the boundary conditions are not correct, if the computational domain is not large enough, etc., the solution can be erroneous and nonphysical no matter how fine the grid.

(*b*) ***True***: Each component of the Navier-Stokes equation is a transport equation.

(*c*) ***True***: The four-sided cells of a 2-D structured grid require less cells than do the triangular cells of a 2-D unstructured grid. (Note however, that some unstructured cells can be four-sided as well as three-sided.)

(*d*) ***True***: Turbulence models are *approximations* of the physics of a turbulent flow, and unfortunately are not universal in their application.

15-45C

Solution We are to discuss right-left symmetry as applied to a CFD simulation and to a potential flow solution.

Analysis In the time-averaged CFD simulation, we are not concerned about top-bottom fluctuations or periodicity. Thus, top-bottom symmetry can be assumed. However, fluid flows do not have upstream-downstream symmetry in general, even if the geometry is perfectly symmetric fore and aft. In the problem at hand for example, the flow in the channel develops downstream. Also, the flow exiting the left channel enters the circular settling chamber like a jet, separating at the sharp corner. At the opposite end, fluid leaves the settling chamber and enters the duct more like an inlet flow, without significant flow separation. **We certainly cannot expect fore-aft symmetry in a flow such as this**.

On the other hand, potential flow of a symmetric geometry yields a symmetric flow, so it would be okay to cut our grid in half, invoking fore-aft symmetry.

Discussion If unsteady or oscillatory effects were important, we should not even specify top-bottom symmetry in this kind of flow field.

15-46C

Solution We are to discuss improvements to the given computational domain.

Analysis (*a*) Since Gerry is not interested in unsteady fluctuations (which may be unsymmetric), he could eliminate half of the domain. In other words, **he could assume that the axis is a plane of symmetry between the top and bottom of the channel**. Gerry's grid would be cut in size by a factor of two, leading to approximately half the required CPU time, but yielding virtually identical results.

(*b*) The fundamental flaw is that **the outflow boundary is not far enough downstream**. There will likely be flow separation at the corners of the sudden contraction. With a duct that is only about three duct heights long, it is possible that there will be reverse flow at the outlet. Even if there is no reverse flow, the duct is nowhere near long enough for the flow to achieve fully developed conditions. Gerry should extend the outlet duct by many duct heights to allow the flow to develop downstream and to avoid possible reverse flow problems.

Discussion The inlet appears to be perhaps too short as well. If Gerry specifies a fully developed channel flow velocity profile at the inlet, his results may be okay, but again it is better to extend the duct many duct heights beyond what Gerry has included in his computational domain.

15-47C

Solution We are to discuss a feature of modern computer systems for which nearly equal size multiblock grids are desirable.

Analysis The fastest computers are multi-processor computers. In other words, the computer system contains more than one CPU – a **parallel computer**. Modern parallel computers may combine 32, 64, 128, or more CPUs or *nodes*, all working together. In such a situation it is natural to let each node operate on one block. If all the nodes are identical (equal speed and equal RAM), the system is most efficient if the blocks are of similar size.

Discussion In such a situation there must be communication between the nodes. At the interface between blocks, for example, information must pass during the CFD iteration process.

15-48C

Solution We are to discuss the difference between multigridding and multiblocking, and we are to discuss how they may be used to speed up a CFD calculation. Then we are to discuss whether multigridding and multiblocking can be applied together.

Analysis *Multigridding* has to do with the resolution of an established grid during CFD calculations. With multigridding, **solutions of the equations of motion are obtained on a coarse grid first, followed by successively finer grids**. This speeds up convergence because the gross features of the flow are quickly established on the coarse grid (which takes less CPU time), and then the iteration process on the finer grid requires less time.

Multiblocking is something totally different. It refers to the **creation of two or more separate blocks or zones, each with its own grid**. The grids from all the blocks collectively create the overall grid. As discussed in the previous problem, multiblocking can have some speed advantages if using a parallel-processing computer. In addition, some CFD calculations would require too much RAM if the entire computational domain were one large block. In such cases, the grid can be split into multiple blocks, and the CFD code works on one block at a time. This requires less RAM, although information from the dormant blocks must be stored on disk or solid state memory chips, and then swapped into and out of the computer's RAM.

There is no reason why multigridding cannot be used on each block separately. Thus, **multigridding and multiblocking can be used together**.

Discussion Although all the swapping in and out requires more CPU time and I/O time, for large grids multiblocking can sometimes mean the difference between being able to run and not being able to run at all.

15-49C

Solution We are to discuss why we should spend a lot of time developing a multiblock structured grid when we could just use an unstructured grid.

Analysis There are several reasons why a structured grid is "better" than an unstructured grid, even for a case in which the CFD code can handle unstructured grids. First of all **the structured grid can be made to have better resolution *with fewer cells* than the unstructured grid**. This is important if computer memory and CPU time are of concern. Depending on the CFD code, the solution **may *converge more rapidly* with a structured grid**, and the results may be **more accurate**. In addition, by creating multiple blocks, we can more easily **cluster cells** in certain blocks and locations where high resolution is necessary, since we have much more control over the final grid with a structured grid.

Discussion As mentioned in this chapter, time spent creating a good grid is usually time well spent.

15-50

Solution We are to calculate flow through a single-stage heat exchanger.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-51

Solution We are to study the effect of heating element angle of attack on heat transfer through a single-stage heat exchanger.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-52

Solution We are to calculate flow through a single-stage heat exchanger.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-53

Solution We are to study the effect of heating element angle of attack on heat transfer through a two-stage heat exchanger.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-54

Solution We are to study the effect of spin on a cylinder using CFD, and in particular, analyze the lift force.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-55

Solution We are to study the effect of spin speed on a spinning cylinder using CFD, and in particular, analyze the lift force as a function of rotational speed in nondimensional variables.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-56

Solution We are to study flow into a slot along a wall using CFD.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.

15-57

Solution We are to calculate laminar flow into a 2-D slot, compare with irrotational flow theory, and with results of the previous problem, and discuss the vorticity field.

Assumptions **1** The flow is steady and 2-D. **2** The flow is laminar.

Analysis The flow field does not change much from the previous problem, except that a thin boundary layer shows up along the floor. The vorticity is confined to a region close to the floor – vorticity is negligibly small everywhere else, so the irrotational flow approximation is appropriate everywhere except close to the floor.

Discussion The irrotational flow approximation is very useful for suction-type flows, as in air pollution control applications (hoods, etc.).

15-58

Solution We are to model the flow of air into a vacuum cleaner using CFD, and we are to compare the results to those obtained with the potential flow approximation.

Analysis We must include a second length scale in the problem, namely the width w of the vacuum nozzle. For the CFD calculations, we set $w = 2.0$ mm and place the inlet plane of the vacuum nozzle at $b = 2.0$ cm above the floor (Fig. 1). Only half of the flow is modeled since we can impose a symmetry boundary condition along the y-axis. We use the same volumetric suction flow rate as in the example problem, i.e., $\dot{V}/L = 0.314$ m^2/s, but in the CFD analysis we specify only *half* of this value since we are modeling half of the flow field.

Results of the CFD calculations are shown in Fig. 2. Fig. 2a shows a view of streamlines in the entire computational plane. Clearly, the streamlines far from the inlet of the nozzle appear as rays into the origin; from "far away" the flow feels the effect of the vacuum nozzle in the same way as it would feel a line sink. In Fig. 2b is shown a close-up view of these same streamlines. Qualitatively, the streamlines appear similar to those predicted by the irrotational flow approximation. In Fig 2c we plot contours of the magnitude of vorticity. Since irrotationality is defined by zero vorticity, these vorticity contours indicate where the irrotational flow approximation is valid – namely in regions where the magnitude of vorticity is negligibly small.

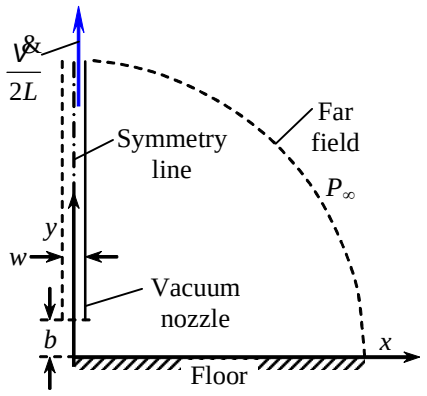
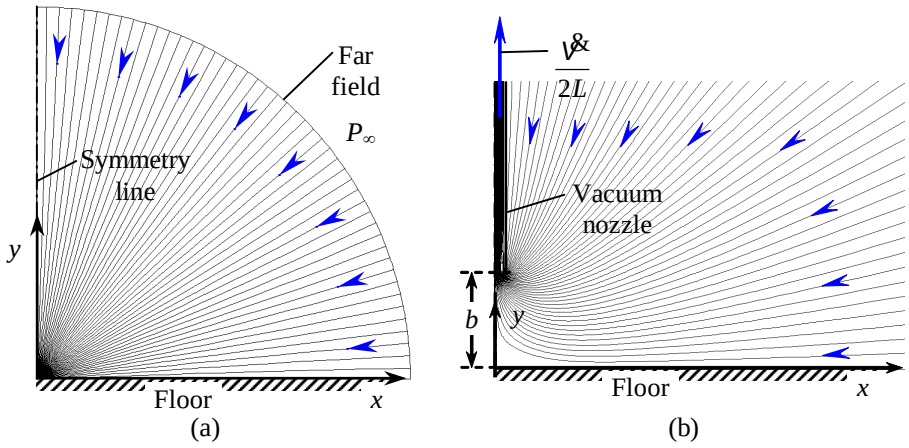


FIGURE 1
CFD model of air being sucked into a vertical vacuum nozzle; the y-axis is a line of symmetry (not to scale –the far field is actually much further away from the nozzle than is sketched here).



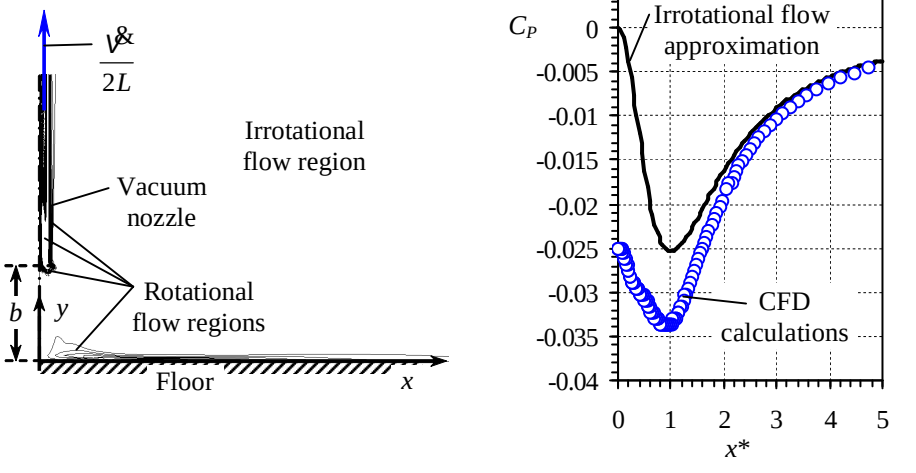


FIGURE 2
CFD calculations of flow into the nozzle of a vacuum cleaner; (a) streamlines in the entire flow domain, (b) close-up view of streamlines, (c) contours of constant magnitude of vorticity illustrating regions where the irrotational flow approximation is valid, and (d) comparison of pressure coefficient with that predicted by the irrotational flow approximation.

We see from Fig. 2c that vorticity is negligibly small everywhere in the flow field

except close to the floor, along the vacuum nozzle wall, near the inlet of the nozzle, and inside the nozzle duct. In these regions, net viscous forces are *not* small and fluid particles *rotate* as they move; the irrotational flow approximation is not valid in these regions. Nevertheless, it appears that the irrotational flow approximation is valid throughout the majority of the flow field. Finally, the pressure coefficient predicted by the irrotational flow approximation is compared to that calculated by CFD in Fig. 2d.

Discussion For x^* greater than about 2, the agreement is excellent. However, the irrotational flow approximation is not very reliable close to the nozzle inlet. Note that the irrotational flow prediction that the minimum pressure occurs at $x^* \approx 1$ is verified by CFD.

15-59

Solution We are to compare CFD calculations of flow into a vacuum cleaner for the case of laminar flow versus the inviscid flow approximation.

Analysis Students will have varied results, depending on the grid generation code, CFD code, turbulence model, and their choice of computational domain, etc.

Discussion Instructors can add more details to the problem statement, if desired, to ensure consistency among the students' results.
