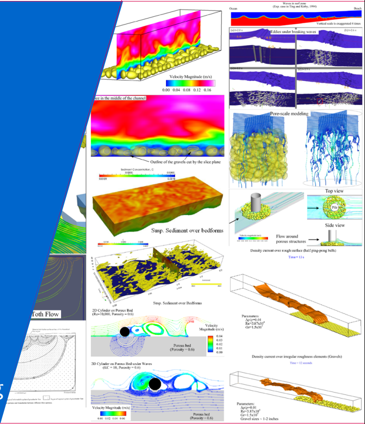
PENNSTATE

1855

CE597D Special Topics, Fall, 2014

Chapter 4, Part 2: Numerical Solution of Advection-Diffusion Equations

Xiaofeng Liu, Ph.D., P.E.
Assistant Professor
Department of Civil and Environmental Engineering
Pennsylvania State University


Velocity Magnitude (m/s)
Outline of the gravels cut by the slice plane
Susp. Sediment over bedforms
Susp. Sediment over Bedforms
Toth Flow
2D Cylinder on Porous Bed
2D Cylinder on Porous Bed under Waves
Porous bed (Porosity = 0.6)
Waves in surf zone
Eddies under breaking waves
Pore-scale modeling
Top view
Side view
Flow around porous structures
Density current over rough surface (half ping-pong balls)
Density current over irregular roughness elements (Gravels)
Gravel sizes ~ 1-2 inches

Outline

Discretization

The full generic, steady, advection-diffusion equation

$$\underbrace{\nabla \cdot (\rho \mathbf{u} \phi)}_{\text{convection term}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{diffusion term}} + \underbrace{S_\phi}_{\text{source term}}$$

Integral form over a control volume (after the application of Gauss's theorem):

$$\int_A \mathbf{n} \cdot (\rho \mathbf{u} \phi)\, dA = \int_A \mathbf{n} \cdot (\Gamma \nabla \phi)\, dA + \int_{CV} S_\phi\, dV$$

which represents the balance in the control volume between the advective flux, diffusive flux, and source generation (or destruction).

Discretization

The full generic, steady, advection-diffusion equation

$$\underbrace{\nabla \cdot (\rho \mathbf{u} \phi)}_{\text{convection term}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{diffusion term}} + \underbrace{S_\phi}_{\text{source term}}$$

Integral form over a control volume (after the application of Gauss's theorem):

$$\int_A \mathbf{n} \cdot (\rho \mathbf{u} \phi) \, dA = \int_A \mathbf{n} \cdot (\Gamma \nabla \phi) \, dA + \int_{CV} S_\phi \, dV$$

which represents the balance in the control volume between the advective flux, diffusive flux, and source generation (or destruction).

We already know how to deal with the diffusive flux and the source term using central difference scheme. The only addition is how to deal with the advective flux.

Discretization

Comparing the diffusive flux and the advective flux, the difference is:

- The diffusion affects the transport process in all directions
- The advection affects the transport process only in the flow direction.

So the numerical schemes for the advective flux need to consider this characteristic.

For steady state problem, this affects the choice on cell size in order to have stable solution.

Discretization

As for the pure diffusion problem, we will use 1D case for demonstration.

We also assume that the velocity field $\mathbf{u}$ is already know from the solution of the Navier-Stokes equation (next chapter).

We further assume these velocities are stored (or somehow can be evaluated) at the faces of the control volumes.

Discretization

As for the pure diffusion problem, we will use 1D case for demonstration.

We also assume that the velocity field $\mathbf{u}$ is already know from the solution of the Navier-Stokes equation (next chapter).

We further assume these velocities are stored (or somehow can be evaluated) at the faces of the control volumes.

We will introduce several different numerical schemes and discuss.

Discretization

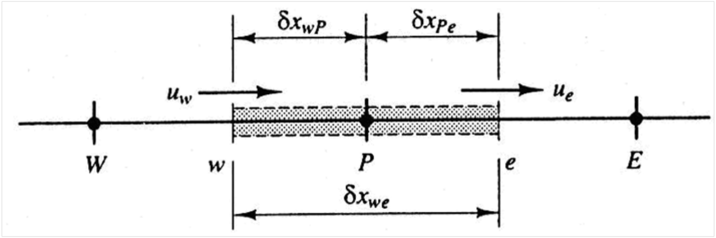
Steady 1D advection-diffusion equation without source

$$\underbrace{\frac{d}{dx}(\rho u \phi)}_{\text{convection term}} = \underbrace{\frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right)}_{\text{diffusion term}}$$

The velocity is from the solution of N-S equations. So from the continuity equation,

$$\frac{d(\rho u)}{dx} = 0$$

Consider a 1D control volume:



Discretization

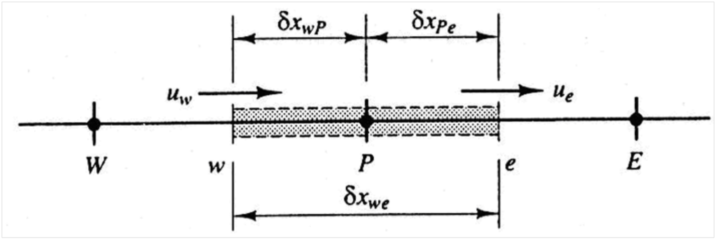
Integration of the governing equation over the control volume, then we have

$$(\rho u A\phi)_e - (\rho u A\phi)_w = \left(\Gamma A \frac{d\phi}{dx}\right)_e - \left(\Gamma A \frac{d\phi}{dx}\right)_w$$

In the above equation, there are four flux terms for each control volume. For convenience, we define two variables F and D :

- advective mass flux per unit area: $F = \rho u$
- diffusion conductance: $D = \frac{\Gamma}{\delta x}$

Assuming the areas A are constant in this 1D problem so they can cancel out.



Discretization

Using the same central difference scheme for the diffusion term, we have

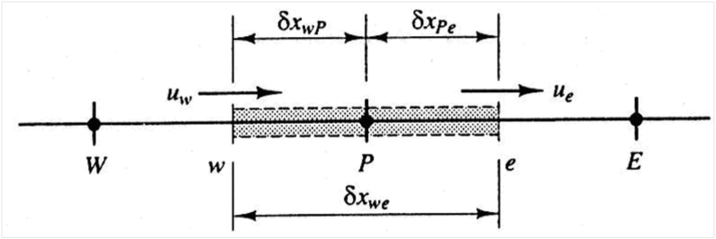
$$F_e\phi_e - F_w\phi_w = D_e\left(\phi_E - \phi_P\right) - D_w\left(\phi_P - \phi_W\right)$$

The integration of the continuity equation gives

$$\left(\rho u A\right)_e - \left(\rho u A\right)_w = 0$$

which can be simply be noted as

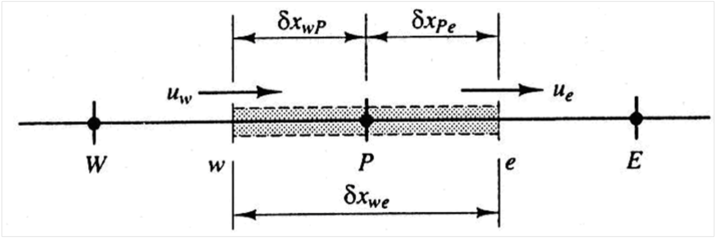
$$F_e - F_w = 0$$



Discretization

To evaluate the fluxes across the faces *e* and *w*, the next step is to interpolate the transported variable ϕ to the faces. We will discuss the following:

- What constitutes a "good" numerical scheme?
- Specific schemes
  - Central differencing
  - Upwind differencing
  - Hybrid differencing
  - Power law
  - QUICK scheme
  - TVD schemes



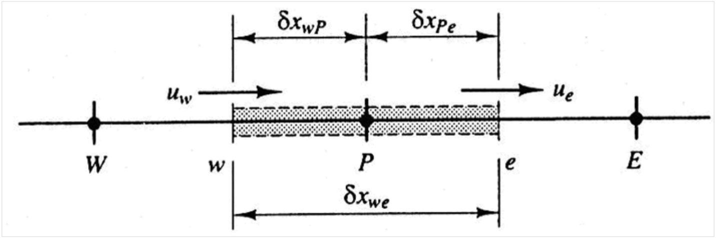
Discretization

In summary, to discretize the advection term $\nabla \cdot (\rho \mathbf{u} \phi)$, we need to specify the interpolation scheme to evaluate the variable at the interface ϕ_e and ϕ_w.
So in `fvSchemes` file, we have

```
Gauss <interpolationScheme>
```

For example:

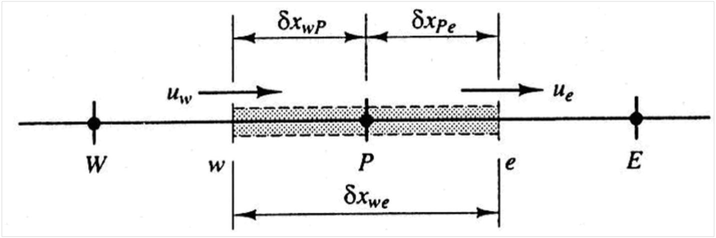
```
div(phi,U) Gauss linear;
```



Discretization

So what constitutes a “good” numerical scheme? Properties of discretization schemes:

- Conservativeness: conservation of the transported property ϕ. In FVM, to ensure this, the flux leaving a control volume through a face should be equal to the flux entering the adjacent cell through the same face.



Discretization

So what constitutes a "good" numerical scheme? Properties of discretization schemes:

- Boundedness: The value of the solution should be bounded, which is usually determined by the diagonal dominance of the resulting matrix A.

  For this reason, the linearization practice of the source $S = S_u + S_p\phi$ should ensure S_p is always negative, which adds to the coefficient $a_P = \sum a_{nb} - S_p$.

  Another essential requirement for boundedness is that all coefficients of the discretized equation should have the same sign (a_P, a_W and a_E), which implies the increase in ϕ at one cell will cause the increase at neighbouring cells.

  $$a_P\phi_P = a_W\phi_W + a_E\phi_E + S_u \tag{1}$$

  If these conditions are not satisfied, solution might not converge or has oscillations. The value of the property ϕ should also be physically meaningful. For example, concentration should not be negative.

Discretization

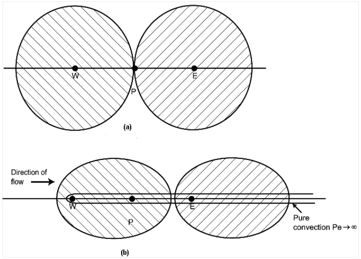
So what constitutes a "good" numerical scheme? Properties of discretization schemes:

- Transportiveness: Define Peclet number (ratio between advection and diffusion)

$$Pe = \frac{F}{D} = \frac{\rho u}{\Gamma/\delta x}$$

  The relationship between the directionality of the influencing and the flow direction and the magnitude of the Pe number.

In this figure, we have two cases: (a) $Pe \to 0$ and (b) $Pe \neq 0$ as well as $Pe \to +\infty$.



Central differencing scheme

Central differencing has been used for the diffusion term. Let's also apply it to the advective term. For a uniform grid, the face values can be written as

$$\phi_e = \frac{\phi_P + \phi_E}{2}, \qquad \phi_w = \frac{\phi_W + \phi_P}{2}$$

Substitute these into the advective flux terms, we have

$$\frac{F_e}{2}\left(\phi_P + \phi_E\right) - \frac{F_w}{2}\left(\phi_W + \phi_P\right) = D_e\left(\phi_E - \phi_P\right) - D_w\left(\phi_P - \phi_W\right)$$

Write this into the familiar format of $a_P\phi_P = a_W\phi_W + a_E\phi_E$, with

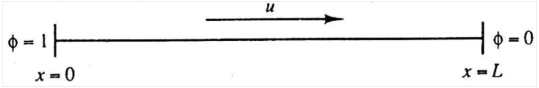
a_W	a_E	a_P
$D_w + \frac{F_w}{2}$	$D_e - \frac{F_e}{2}$	$D_w + D_e - \frac{F_w}{2} + \frac{F_e}{2}$

You can compare these coefficients with those for pure diffusion without source terms. The only difference is the term from the advection (F_e and F_w).

Central differencing scheme

Example: 1D advection-diffusion without source. B.C.s are fixed values at both ends.

$$\phi(x=0)=1, \qquad \phi(x=L)=0$$



The analytical solution for this:

$$\frac{\phi-\phi_0}{\phi_L-\phi_0}=\frac{\exp\left(\rho u x/\Gamma\right)-1}{\exp\left(\rho u L/\Gamma\right)-1}$$

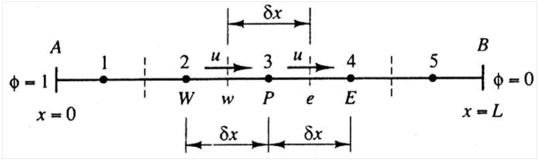
We consider three cases:

- Case 1: $u = 1\ m/s$, with 5 cells
- Case 2: $u = 2.5\ m/s$, with 5 cells
- Case 3: $u = 2.5\ m/s$, with 20 cells

with $L = 1.0\ m$, $\rho = 1.0\ kg/m^3$, $\Gamma = 0.1\ kg/(m \cdot s)$.

Central differencing scheme

We will use 5 cells first. For the three internal cells, the discretization equation has been derived. We need to work on the boundary cells.



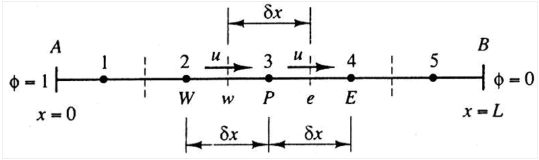
Let's look at boundary cell 1. The integration of the governing equation around cell 1 uses central differencing for both advection and diffusion. The value of ϕ at the west face (boundary A) is given. No need to interpolate!

This yields the equation for cell 1:

$$\frac{F_e}{2}\left(\phi_P + \phi_E\right) - \underbrace{F_A\phi_A}_{\text{from B.C. at A}} = D_e\left(\phi_E - \phi_P\right) - \underbrace{D_A\left(\phi_P - \phi_A\right)}_{\text{from B.C. at A}}$$

where $D_A = \frac{D}{\delta x/2} = \frac{2D}{\delta x} = 2D$.

Central differencing scheme



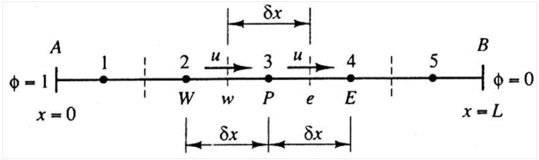
Do the same for boundary cell 5:

$$\underbrace{F_B\phi_B}_{\text{from B.C. at B}} - \frac{F_w}{2}(\phi_P + \phi_W) = \underbrace{D_B(\phi_B - \phi_P)}_{\text{from B.C. at B}} - D_w(\phi_P - \phi_W)$$

where $D_B = \frac{D}{\delta x/2} = \frac{2D}{\delta x} = 2D$.

We can say the similar observation on the effect of the boundary condition. The link to the boundary side has been suppressed and the boundary flux is entered through the source term.

Central differencing scheme



The above process yields one discretized equation for each cell. So there will be a total of 5 equations.

(i) Case 1: $u = 1\ m/s$

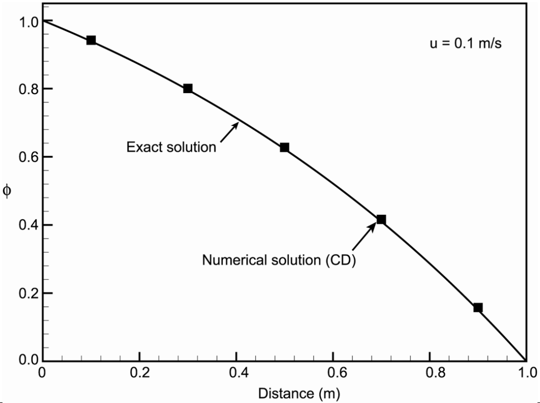
Plugging the values of the parameters, we have the following set of algebraic equations

$$\begin{bmatrix} 1.55 & -0.45 & 0 & 0 & 0 \\ -0.55 & 1.0 & -0.45 & 0 & 0 \\ 0 & -0.55 & 1.0 & -0.45 & 0 \\ 0 & 0 & -0.55 & 1.0 & -0.45 \\ 0 & 0 & 0 & -0.55 & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} 1.1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

Central differencing scheme

Solving the set of algebraic equations (more on how to solve it later), we have the solutions

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} 0.9421 \\ 0.8006 \\ 0.6276 \\ 0.4163 \\ 0.1579 \end{bmatrix} \tag{3}$$



Central differencing scheme

This example has been implemented in OpenFOAM® and the solver name is *advectionDiffusionFoam*.

```
fvScalarMatrix TEqn
(
     fvm::ddt(rho, T)
   + fvm::div(rho*phi, T)
   - fvm::laplacian(DT, T)
);

//save the matrix and source to a file
saveMatrix(TEqn);
```

Don't worry about the "fvm::ddt(rho, T)" part. It can be turned off to solve for steady state.

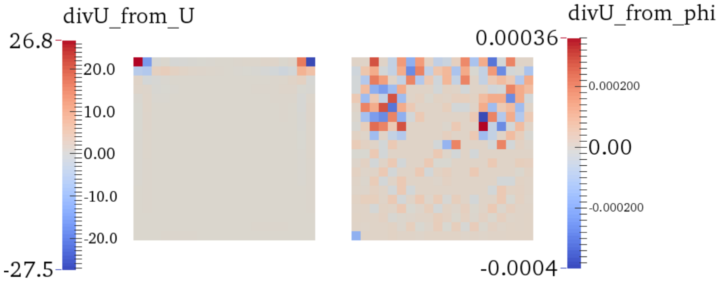
Central differencing scheme

The code needs to generate the advective flux $phi = uA$. So the user needs to provide a velocity field u or a flux field phi. It is done in the included file `createPhi.H`:

```
surfaceScalarField phi
(
    IOobject
    (
        "phi",
        runTime.timeName(),
        mesh,
        IOobject::READ_IF_PRESENT,
        IOobject::AUTO_WRITE
    ),
    linearInterpolate(U) & mesh.Sf()
);
```

Central differencing scheme

The use of `surfaceScalarField phi` field from the solution of Navier-Stokes equations is recommended because only `phi` is strictly divergence free.



Divergence of the cavity driven flow case.

A tool called `divU` is written to calculate the divergence from both `U` and `phi`.

$$\nabla \cdot (\mathrm{u}T) = \underbrace{T\nabla \cdot \mathrm{u}}_{\text{extra source term if u is not divergence free}} + \mathrm{u} \cdot (\nabla T)$$

Central differencing scheme

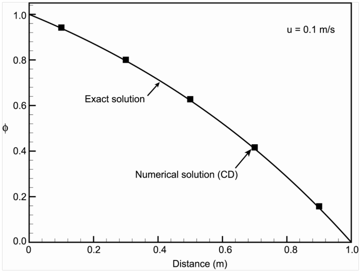
This example case is called *case1*. The setup is the same as Example 5.1. Note the following setup in *system/fvSchemes* file:

- *steadyState* for the *ddtSchemes* entry
- *Gauss linear* for the *div((rho*phi), T)* entry: equivalent of central difference scheme for advective flux.
- *Gauss linear corrected* for the *laplacian(DT, T)* entry: equivalent of central difference scheme for diffusive flux.

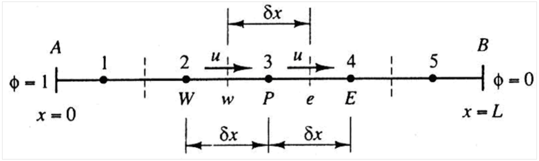
Central differencing scheme

In the initial condition directory "0", there is a file "U", where you need to specify the advection velocity. In this Case 1, $u = 0.1\ m/s$.
The parameters are setup in *constant/transportProperties* file:

```
DT              DT [ 1 -1 -1 0 0 0 0 ] 0.1;
rho            rho [ 1 -3 0 0 0 0 0 ] 1;
```



Central differencing scheme



(i) Case 2: $u = 2.5\ m/s$

Plugging the values of the parameters, we have the following set of algebraic equations

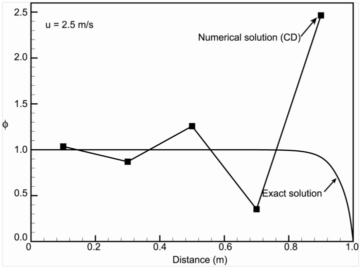
$$\begin{bmatrix} 2.75 & 0.75 & 0 & 0 & 0 \\ -1.75 & 1.0 & 0.75 & 0 & 0 \\ 0 & -1.75 & 1.0 & 0.75 & 0 \\ 0 & 0 & -1.75 & 1.0 & 0.75 \\ 0 & 0 & 0 & -1.75 & 0.25 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

In OpenFOAM, the solver is still *advectionDiffusionFoam*. The case is called *case2*. The only place you need to make modification is the the "U" velocity file. Change the advective velocity to 2.5 m/s.

Central differencing scheme

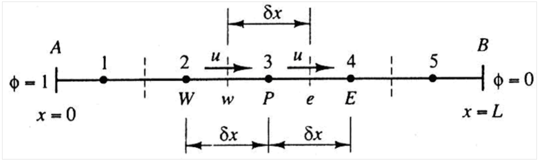
Solving the set of algebraic equations, we have the solutions

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} 1.03563 \\ 0.869355 \\ 1.25733 \\ 0.352053 \\ 2.46437 \end{bmatrix} \tag{5}$$



The solution is really bad. It has severe oscillations.

Central differencing scheme

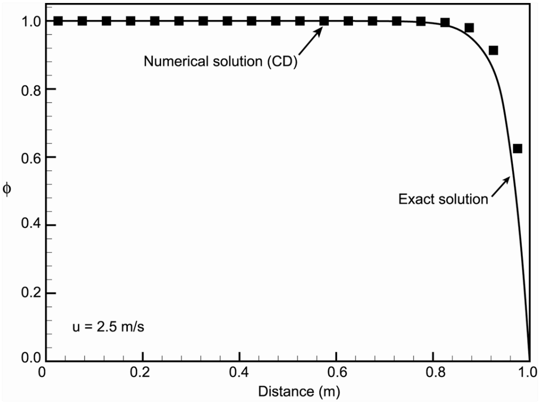


(i) Case 3: $u = 2.5\ m/s$ with 20 cells

Plugging the values of the parameters, we have the following set of algebraic equations. The linear system $Ax = b$ is so big. It will be omitted here.
In OpenFOAM, the solver is still *advectionDiffusionFoam*. The case is called *case3*. You need to modify the "blockMeshDict" file to have 20 cells in the x direction.

Central differencing scheme

Solving the set of algebraic equations to get the solutions



The solution with refined mesh is now good! The reason for this behavior will be discussed later.

Central differencing scheme

Now let's evaluate the central differencing scheme based on our criteria of "good" scheme.

- Conservativeness: As you can see the fluxes (both advective and diffusive) evaluated at faces using central differencing are consistent. So it is conservative.
- Boundedness: The discretized equation for internal cells has the form of $a_P \phi_P = a_W \phi_W + a_E \phi_E$, with

a_W	a_E	a_P
$D_w + \frac{F_w}{2}$	$D_e - \frac{F_e}{2}$	$D_w + D_e - \frac{F_w}{2} + \frac{F_e}{2}$

  For the simple 1D case, the continuity equation states that $F_e - F_w = 0$. Then the coefficient $a_P = a_W + a_E$. This satisfies the "weak" diagonal dominant condition for stable solution.

Central differencing scheme

- Boundedness: (continue)
  For the east face, the coefficient $a_E = D_e - F_e/2$. So the advective contribution is negative. If advection is dominant, then a_E could be negative. For a_E to remain positive, the condition is

$$\frac{F_e}{D_e} = Pe_e < 2$$

  If $Pe_e > 2$, a_E is negative and this violates one of the requirements for boundedness.

  In the example, $Pe = 5$ in Case 2, which produces a solution with oscillations. $Pe = 0.2$ in Case 1 and $Pe = 1.25$ in Case 3, which produce good results.

Central differencing scheme

In practice, to make $Pe = \frac{\rho u \delta x}{\Gamma} < 2$, you will have the following options:

- The flow is diffusion dominant with low Reynolds number. Very rare for practical purpose.
- Reduce the grid size (δx).

Both are not favorable for general purpose computation. That is the reason we need to have other schemes such as upwind, hybrid, power-law, QUICK and TVD.

Central differencing scheme

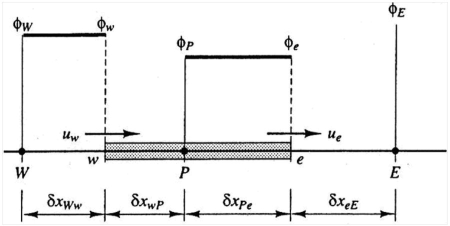
- Transportiveness: The central differencing scheme does not recognize the direction of the flow. It simply use the average value on both sides of the faces. So it does not have the transportiveness.
- Accuracy: Second-order according the Taylor series truncation error analysis.

Upwind differencing scheme

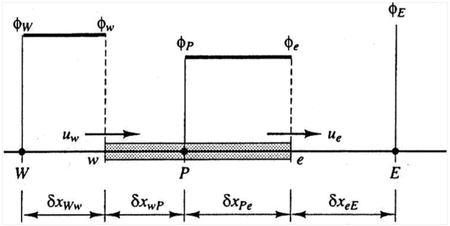
The idea: to consider the direction of the flow and its influence on the interpolation of face value.

In upwind scheme, the influence of the flow direction is taken to the extreme, i.e., the convected value of ϕ at a cell face is taken to be equal to the upstream cell center value.

For example in this figure, the flow is in the positive x direction. So the upstream cells of faces w and e are W and P, respectively.



Upwind differencing scheme



The discretization equation now should be

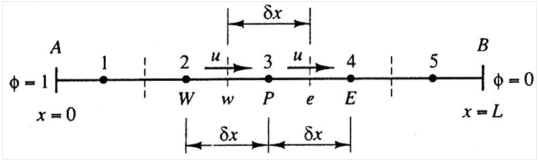
$$F_e\phi_P - F_w\phi_W = D_e\left(\phi_E - \phi_P\right) - D_w\left(\phi_P - \phi_W\right)$$

$$\left(D_w + D_e + F_e\right)\phi_P = \left(D_w + F_w\right)\phi_W + D_e\phi_E$$

Similar equation can be derived for flow in the negative direction.

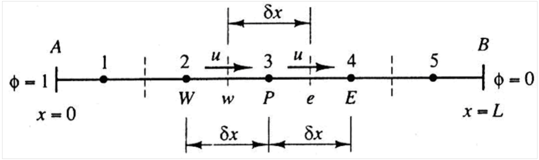
Upwind differencing scheme

We will use UD to solve the same problem in Example 5.1 for (i) $u = 0.1$ m/s, (ii) $u = 2.5$ m/s.



Flow is in the positive x direction. For the internal nodes 2, 3, and 4, the equation derived in the previous slide can be used.

Upwind differencing scheme



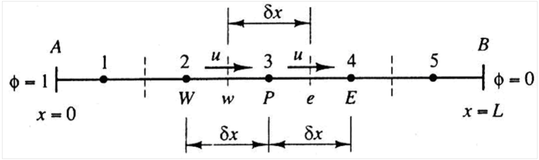
At the boundary, it is different. For boundary cell 1, the use of UD for the convective term gives

$$F_e \phi_P - F_A \phi_A = D_e \left(\phi_E - \phi_P\right) - D_A \left(\phi_P - \phi_A\right)$$

and for boundary cell 5,

$$F_B \phi_P - F_w \phi_W = D_B \left(\phi_B - \phi_P\right) - D_w \left(\phi_P - \phi_W\right)$$

Upwind differencing scheme

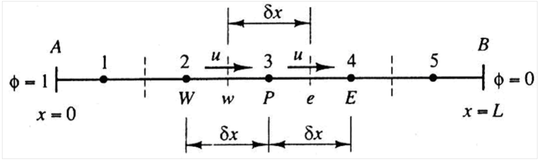


For boundary cell 5,

$$F_B\phi_P - F_w\phi_W = D_B\left(\phi_B - \phi_P\right) - D_w\left(\phi_P - \phi_W\right)$$

There is a debate on how to treat the downstream boundary cell 5 if the B.C. at boundary B is a fixed value type. There are two opinions. One is to strictly follow the upwind differencing idea, the flux should be calculated as the previous equation, i.e. $F_B\phi_P$ because cell center P is the the upstream value for the boundary B.
Another option is to strictly enforcing the fixed value B.C. at boundary B. As such, the flux through the boundary B should be $F_B\phi_B$. This is what OpenFOAM does. There is a long discussion in cfd-online forum.

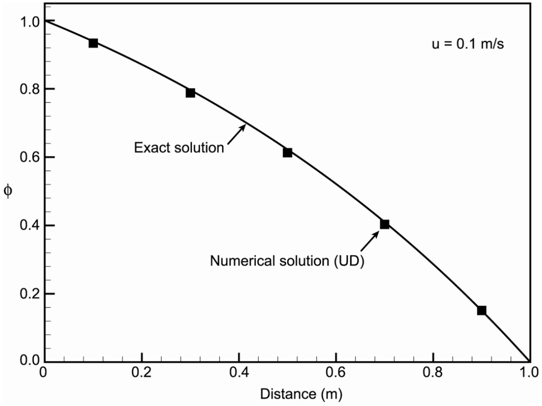
Upwind differencing scheme



Regardless, at the boundary, we have $D_A = D_B = 2\Gamma/\delta x = 2D$ and $F_A = F_B = F$ and the assembly of the matrix and its solution is similar as before.

Upwind differencing scheme

Case 1: $u = 0.1\ m/s$, $F = \rho u = 0.1$, $D = \Gamma/\delta x = 0.1/0.2 = 0.5$, so $Pe = F/D$ $= 0.2$.



The upwind scheme produces good result.

Upwind differencing scheme

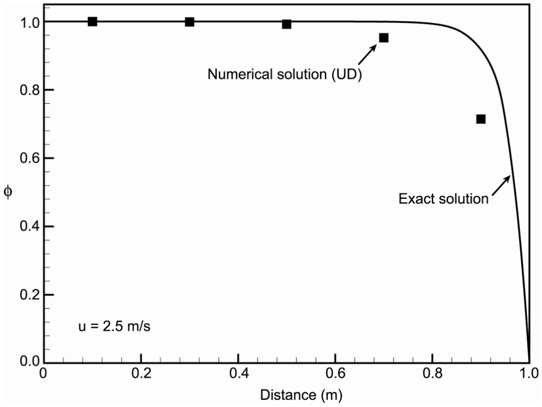
In OpenFOAM, the solver is still *advectionDiffusionFoam*. The case setup is the same as *case1* in the central differencing example. The only place you need to make modification is in *fvSchemes* to have upwind scheme for the advection term:

```
divSchemes
{
   default                  none;
   div((rho*phi),T)         Gauss upwind;
}
```

Due to the difference in how to treat the fix value downstream boundary, the result from OpenFOAM is slightly different for this example. However, in the next example, the result is dramatically different.

Upwind differencing scheme

Case 2: $u = 2.5$ m/s, $F = \rho u = 2.5$, $D = \Gamma/\delta x = 0.1/0.2 = 0.5$, so $Pe = F/D = 5$.



The upwind scheme produces reasonable result if you follow the way in the book. It has some difficulty to produce good result near boundary B. However, OpenFOAM solver produces a diverged solution (Demo on the cluster).

Upwind differencing scheme

Brief assessment of the upwind differencing scheme:

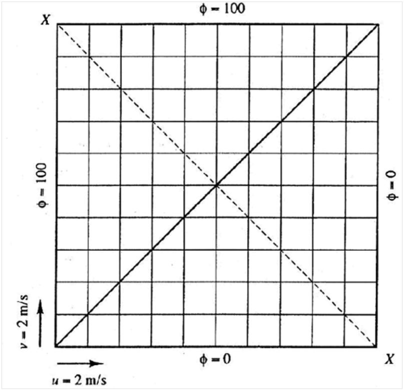
- Conservativeness: it is conservative since the fluxes are calculated in consistent way.
- Boundedness: it is bounded. If the flow velocity field satisfies continuity (usually very strictly required, one reason is here), $F_e - F_w$ in the a_P formula is zero, such that $a_P = a_W + a_E$, which is desirable.
- Transportiveness: the scheme accounts for the direction of the flow, so transportiveness is autmotatically satisfied.
- Accuracy: Only first-order accurate (Appendix A for the Taylor series analysis).

Upwind differencing scheme

One major drawback of the upwind differencing scheme is the false diffusion. When the velocity is not aligned with the grid line, the situation gets worse. To demonstrate, a pure convection process is considered ($\Gamma = 0$).

The analytical solution should be such that all cells above the diagonal line have a value of 100 and all cells below be 0. If we plot the value of ϕ along the $X - X$ line, the analytical solution should be a step function.

Demonstration of the problem on cluster.

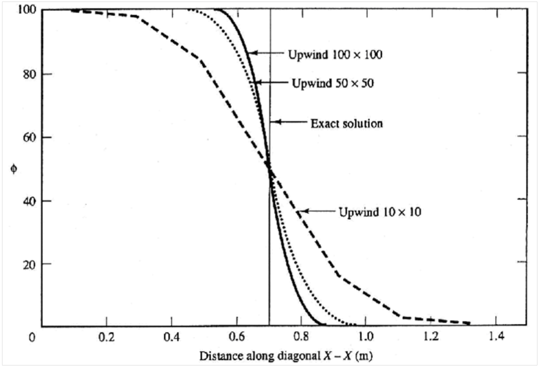


Upwind differencing scheme

Observations of the results using UD with different resolutions:

- The error is the largest for the coarsest mesh.
- Refine the mesh can somewhat reduce the false diffusion.

In practice, UD is not a good choice unless you have no other choices.
For some complicated cases, however, you can use UD to get first-order solution as your initial condition to start high-order simulations.



Hybrid differencing scheme

So far we have:

- Central differencing: second-order accurate but can only be used for $Pe < 2$
- Upwind differencing: first-order accurate but accounting for the transportiveness and can be used for $Pe \geq 2$.

How about we combine both? This is the idea of hybrid differencing scheme proposed in Spalding (1972).

Hybrid differencing scheme

The hybrid differencing scheme relies on local Peclet number to evaluate the flux through interfaces. For example, the local Peclet number on the west face can be defined as:

$$Pe_w = \frac{F_w}{D_w} = \frac{(\rho u)_w}{\Gamma_w / \delta x_{WP}}$$

The hybrid differencing scheme for the net flux per unit area through the west face is as follows:

$$q_w = F_w \left[\frac{1}{2} \left(1 + \frac{2}{Pe_w} \right) \phi_W + \frac{1}{2} \left(1 - \frac{2}{Pe_P} \right) \phi_P \right] \qquad \text{for} \quad -2 < Pe_w < 2$$

$$q_w = F_w \phi_W \qquad \text{for } Pe_w \geq 2$$

$$q_w = F_w \phi_P \qquad \text{for } Pe_w \leq -2$$

It is easy to see that for low *Pe* flow, it is equivalent to central differencing. For $|Pe| > 2$, it is equivalent to upwinding for convection and set the diffusion to zero.

Other differencing schemes

Now it is a good time to list the scheme available in OpenFOAM® .

Scheme	Numerical behaviour
linear	Second order, unbounded
skewLinear	Second order, (more) unbounded, skewness c tion
cubicCorrected	Fourth order, unbounded
upwind	First order, bounded
linearUpwind	First/second order, bounded
QUICK	First/second order, bounded
TVD schemes	First/second order, bounded
SFCD (Self-filtered central differencing)	Second order, bounded
NVD schemes	First/second order, bounded

It is a surprise that the hybrid and later on the power-law scheme are not implemented. It should not be too difficult to implement by yourself. It is a good term project.

Other differencing schemes

Among the available schemes listed above, the closest to hybrid might be the *linearUpwind* scheme.

In the OpenFOAM document, the *linearUpwind* is an interpolation scheme class derived from upwind and returns upwind weighting factors and also applies a gradient-based explicit correction.

Demonstration of *linearUpwind* on cluster.

hybrid differencing scheme

Brief assessment of the hybrid differencing scheme

- Fully conservative
- Unconditionally bounded
- Satisfies transportiveness
- Disadvantage: only first-order

Power-law differencing scheme

Power-law scheme proposed in Patankar (1980) uses the exact solution and produces better results than the hybrid scheme.
The power-law scheme interpolates the face value of a variable, ϕ, using the exact solution to a one-dimensional convection-diffusion equation given below:

$$\frac{\partial}{\partial x}(\rho u \phi) = \frac{\partial}{\partial x}\Gamma\frac{\partial \phi}{\partial x}$$

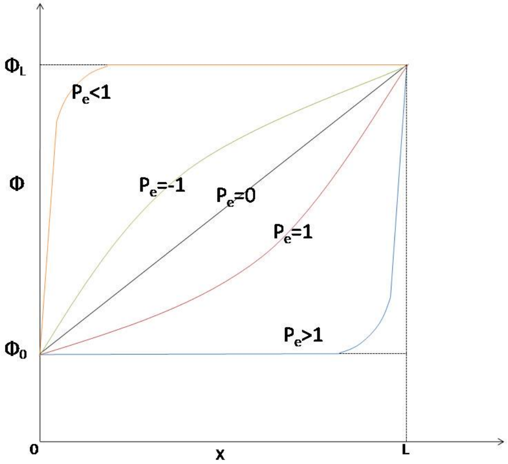
with boundary conditions,

$$\phi_0 = \phi|_{(x=0)} \qquad \phi_L = \phi|_{(x=L)}$$

Power-law differencing scheme

Variation of face value with distance, x is given by the expression,

$$\frac{\phi(x) - \phi_0}{\phi_L - \phi_0} = \frac{\exp(P_e \frac{x}{L})}{\exp(P_e) - 1}$$



Power-law differencing scheme

In the power-law scheme, the flux is evaluated as

$$q_w = F_w \left[\phi_W - \beta_w \left(\phi_P - \phi_W \right) \right] \quad \mathit{for} \quad 0 < Pe < 10$$

where

$$\beta_w = \left(1 - 0.1 Pe_w \right)^5 / Pe_w$$

and

$$q_w = F_w \phi_W \quad \mathit{for} \quad Pe > 10$$

Hybrid and power-law differencing schemes

Interesting reading on the hybrid and power-law and related schemes:

Why you should not use Hybrid, Power-Law or related exponential schemes for convective modelling: there are much better alternatives, International Journal for Numerical Methods in Fluids. Special Issue: Numerical Methods for Thermal Problems. Volume 20, Issue 6, pages 421ÍC442, 30 March 1995

Higher-order differencing scheme

The first-order UD and hybrid schemes ensure stable solution but with sever numerical diffusion. CD is second-order accurate but with unstable solution when $Pe > 2$.

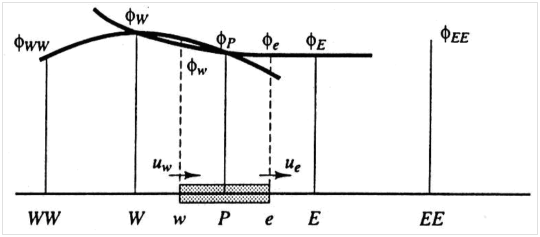
Higher-order schemes involve more neighbour points and reduce the discretization error by using a much bigger numerical stencil.

In this section, we will introduce several higher-order schemes: QUICK (original and improved versions), TVD schemes.

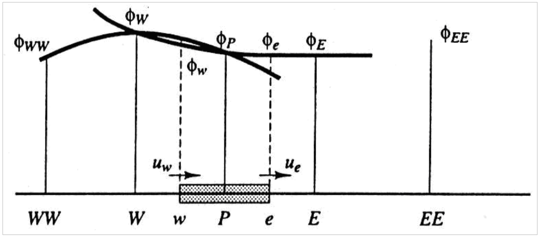
QUICK scheme

Leonard's 1979 original quadratic upwind differencing scheme: quadratic upstream interpolation for convective kinetics (QUICK).

It uses a three-point upstream-weighted quadratic interpolation for cell face values. The three points are the two bracketing nodes on each side of the face and a node on the upstream side.



QUICK scheme



In this figure, if $u_w > 0$ and $u_e > 0$, then a quadratic fit through WW, W, and P is used to evaluate ϕ_w. For ϕ_e, a quadratic fit through W, P, and E is used. For a face between nodes i and $i-1$, with upstream node $i-2$, the QUICK interpolation scheme is given by

$$\phi_{face} = \frac{6}{8}\phi_{i-1} + \frac{3}{8}\phi_i - \frac{1}{8}\phi_{i-2}$$

Correspondingly, the diffusion term can be evaluated using the gradient of the approximating parabola. For uniform grid, it is interesting to note that QUICK scheme is equivalent to CD scheme.

QUICK scheme

If $F_w > 0$ and $F_e > 0$, the QUICK scheme treatment of both the advective and diffusive terms gives the following discretization equation:

$$\left[D_w - \frac{3}{8}F_w + D_e + \frac{6}{8}F_e\right]\phi_P = \left[D_w + \frac{6}{8}F_w + \frac{1}{8}F_e\right]\phi_W + \left[D_e - \frac{3}{8}F_e\right]\phi_E - \frac{1}{8}F_w\phi_W$$

Similar equation can be derived for $F_w < 0$ and $F_e < 0$

QUICK scheme

Brief assessment of the QUICK scheme:

- Conservativeness: The QUICK scheme is conservative as it uses consistent quadratic functions to calculate the control volume face flux.
- Transportiveness: As with the upwind scheme, the transportiveness is present in the QUICK scheme by construction.
- Boundedness: If the flow satisfies the continuity equation, then $a_P = a_W + a_E$; this satisfies the necessary condition which is desirable for boundedness. However, the QUICK scheme have stability problems restricted by the Peclet number. For $F_w < 0$ and $F_e < 0$ the coefficient a_E will be negative when the Peclet number is larger than

  $$Pe = \frac{F_e}{D_e} > \frac{8}{3}$$

  which comes from $a_E = D_e - \frac{3}{8}F_e < 0$. This makes the QUICK scheme conditionally stable.
- Numerical accuracy: Taylor expansion shows that the QUICK scheme is of third order accuracy.

QUICK scheme

In general, the QUICK scheme has third order accuracy and has transportive properties. The solutions are generally low in false diffusion and can produce good results on coarse grids. However, the Peclet number restriction makes this scheme prone to numerical instabilities in complex flows, which can result in overshoots and undershoots.

The original QUICK scheme has been reformulated to improve its stability, for example Hayase et al. (1992). The central idea is called deferred correction with iterations to make the main coefficients positive and satisfy the requirements for conservativeness, boundedness, and transportiveness.

Demo of the QUICK scheme on cluster.

TVD scheme

Higher-order schemes (3rd-order and up) such as QUICK produce undershoot and overshoot (oscillations) when *Pe* is high. For example, if you use QUICK to solve turbulence model equations, it could give unphysical negative values and instability.

TVD: second-order scheme (total variation diminishing) to achieve oscillation-free solution.

In TVD schemes, the tendency towards oscillation is counteracted by adding an artificial diffusion or by adding a weighting towards upstream contribution. In early literature, TVD is also called flux corrected transport (FTC) schemes.

TVD scheme

Assuming velocity $u > 0$, i.e. in the positive x direction. The upwind differencing scheme for the east face value

$$\phi_e = \phi_P$$

Linear upwind differencing (LUD) scheme, a second-order extension of the UD scheme with a correction based on upstream-biased estimate of the gradient:

$$\phi_e = \phi_P + \frac{(\phi_P - \phi_W)}{\delta x}\frac{\delta x}{2} = \phi_P + \frac{(\phi_P - \phi_W)}{2}$$

Similarly, the QUICK scheme can be re-arranged as

$$\phi_e = \phi_P + \frac{1}{8}[3\phi_E - 2\phi_P - \phi_W]$$

and the central differencing scheme can be written as

$$\phi_e = \phi_P + \frac{1}{2}(\phi_E - \phi_P)$$

All above can be written in a general form as

$$\phi_e = \phi_P + \frac{1}{2}\Psi(r)(\phi_E - \phi_P)$$

where Ψ is an appropriate function called flux limiter function.

TVD scheme

If we define the ratio of the upwind-side gradient to downwind-side gradient r

$$r = \frac{\phi_P - \phi_W}{\phi_E - \phi_P}$$

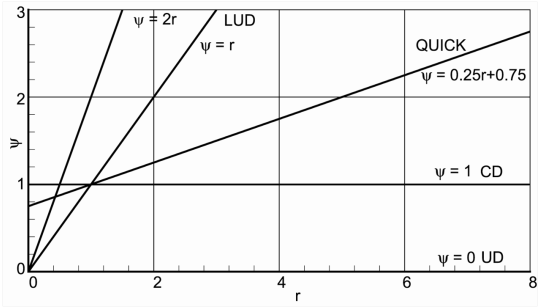
then

For the UD scheme $\Psi(r) = 0$

For the CD scheme $\Psi(r) = 1$

For the LUD scheme $\Psi(r) = r$

For the QUICK scheme $\Psi(r) = (3 + r)/4$



TVD scheme

Total variation for a conserved property ϕ is defined as

$$TV(\phi^n) = \sum_k |\phi_k^n - \phi_{k-1}^n|$$

The desirable property of a stable, high-order and non-oscillating discretization scheme is monotonicity preserving. It does not create local extrema.

For monotonicity to be satisfied, the TV must not increase. For unsteady problems, the TV should diminish with time for monotonicity-preserving schemes, i.e.

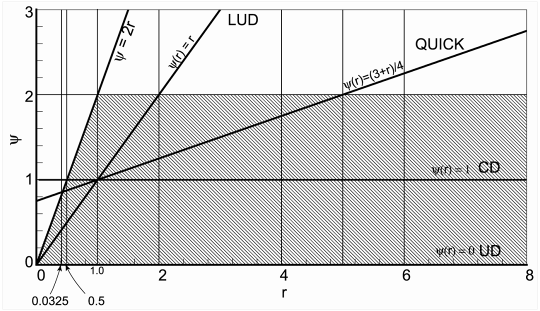
$$TV(\phi^{n+1}) \leq TV(\phi^n).$$

That is where the name TVD comes from.

TVD scheme

Sweby (1984) criteria for TVD schemes: necessary and sufficient conditions for a scheme to be TVD in terms of the $r - \Psi$ relationship.

- If $0 < r < 1$, the upper limit is $\Psi(r) = 2r$, so for TVD schemes $\Psi(r) \leq 2r$
- If $r \geq 1$, the upper limit is $\Psi(r) = 2$, so for TVD schemes $\Psi(r) \leq 2$



Except for UD, all the other schemes are outside of the TVD region for certain value of r.

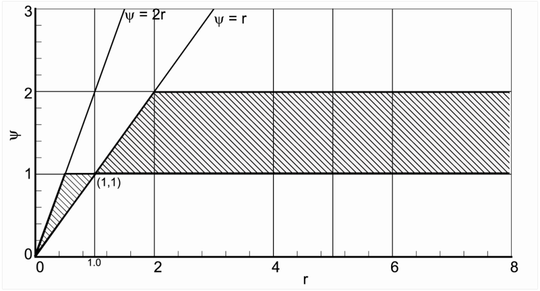
The idea of designing a TVD scheme would be introduce some modifications so as to force the $r - \Psi$ relationship to remain within the shaded area. That is why the function $\Psi(r)$ is called flux limiter function.

TVD scheme

Sweby (1984) also introduced the requirement for second-order accuracy:

- The flux limiter function should pass through the point (1, 1) in the $r - \Psi$ diagram.

Sweby (1984) also showed that the range of possible second-order schemes is bounded by CD and LUD schemes.



TVD scheme

Sweby (1984) finally introduced the symmetry property for limiter functions:

$$\frac{\Psi(r)}{r} = \Psi\left(\frac{1}{r}\right)$$

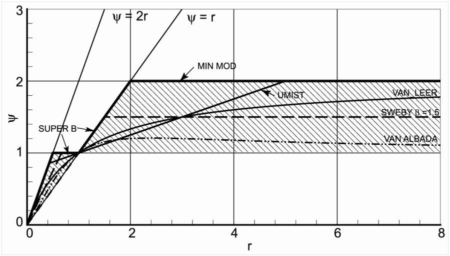
A limiter function which satisfies the symmetry property ensures that the backward- and forward-facing gradients are treated in the same fashion without the need for special coding.

TVD scheme

Flux limiter functions:

Name	$\Psi(r)$	Limiting value	Reference
Van Leer	$\frac{r+\lvert r\rvert}{1+r}$	$\lim_{r\to\infty}\Psi(r)=2$	Van Leer (1974)
Van Albada	$\frac{r+r^2}{1+r^2}$	$\lim_{r\to\infty}\Psi(r)=1$	Van Albada et al. (1982)
Minmod	$\max[0,\ \min[r,\ 1]]$	$\lim_{r\to\infty}\Psi(r)=1$	Roe (1985)
SUPERBEE	$\max[0,\ \min(2r,\ 1),\ \min(r,\ 2)]$	$\lim_{r\to\infty}\Psi(r)=2$	Roe (1985)
QUICK	$\max[0,\ \min(2r,\ (3+r)/4,2)]$	$\lim_{r\to\infty}\Psi(r)=2$	Leonard (1988)
UMIST	$\max[0,\ \min(2r,\ (1+3r)/4,\ (3+r)/4,2)]$	$\lim_{r\to\infty}\Psi(r)=2$	Lien and Leschziner (1993)

In OpenFOAM:
src/finiteVolume/interpolation/surfaceInterpolation/limitedSchemes.



TVD scheme

Brief assessment of the TVD scheme:

- TVD schemes satisfies by default conservativeness, transportiveness and boundedness as they are generalisations of existing schemes with these properties.
- Numerical accuracy: TVD schemes that follows the limitation on the flux limiter are of second order accuracy.

General behaviour:

- Solutions produced by TVD schemes are in general oscillation free, low in false diffusion and of second order.
- Complex limiter functions can require more CPU time than regular differencing schemes.
- When selecting a TVD scheme to use, there are no arguments for one over the other, the performance difference is negligible.

TVD scheme

Demo of TVD scheme.

Questions?