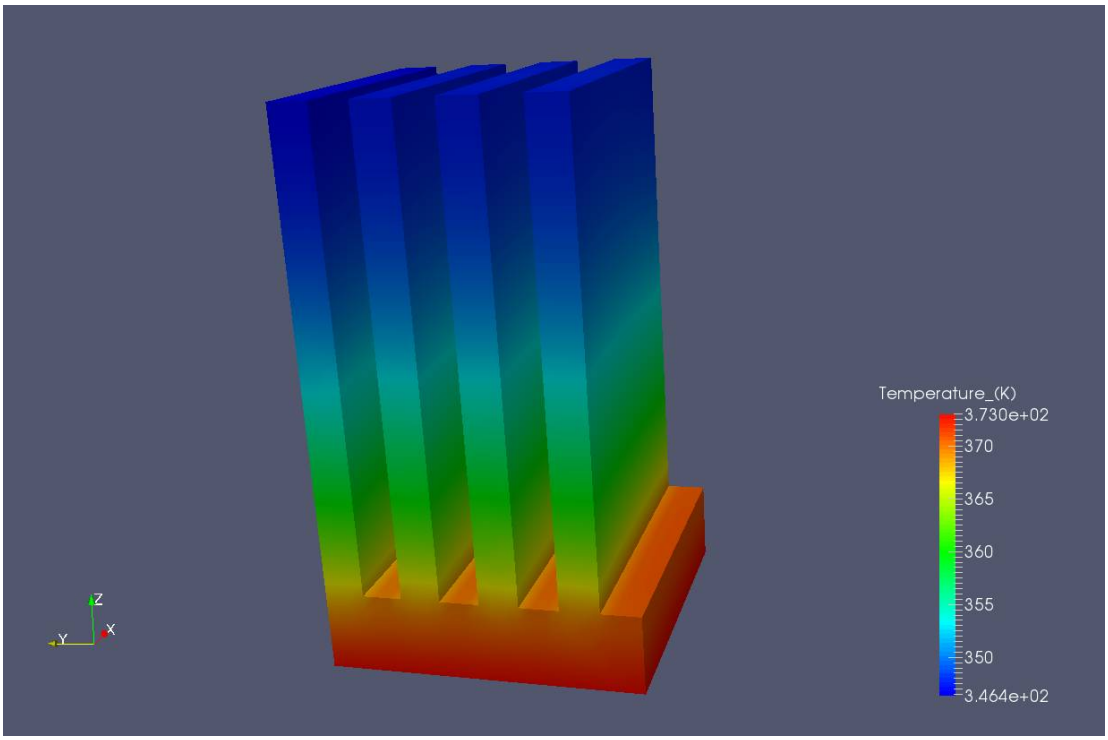
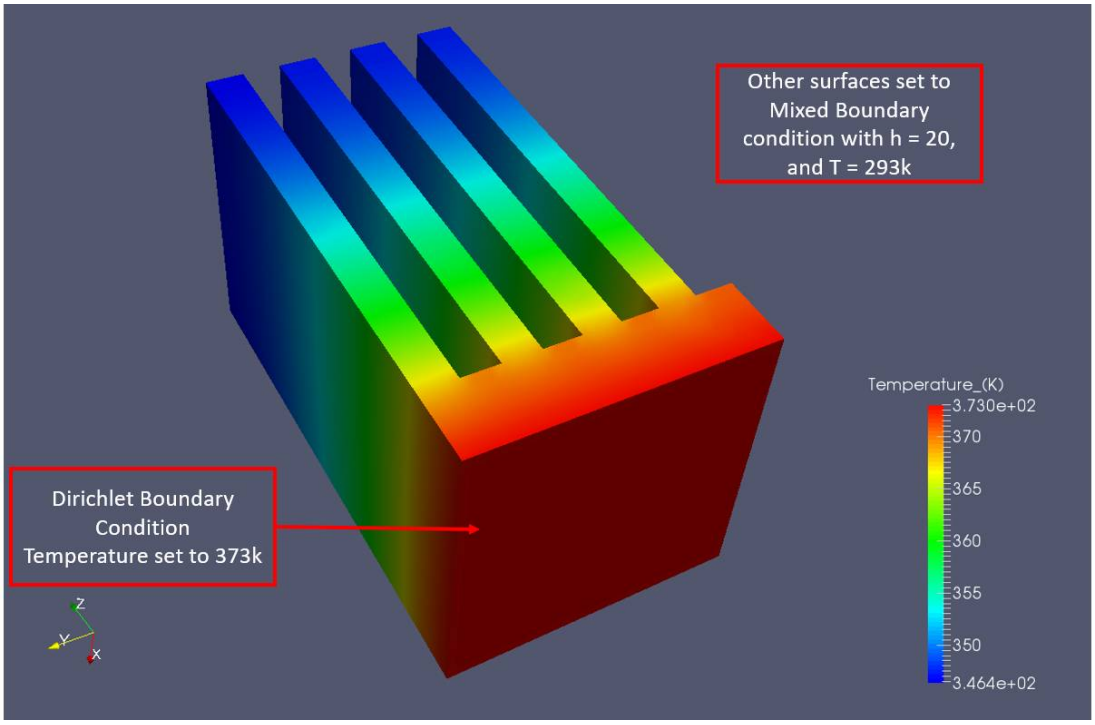
USD is a Finite Volume solver for the Diffusion equation for hexahedral unstructured grids (gmte format). The following is a solution of temperature distribution in a simple heat sink model using USD. The grid file for this model is attached in the repository files.





Using the solver:

The gmte grid file can be created using ICEM CFD grid generator, the grid file should be named test.1.gmte.c and placed in the debug folder.

The Boundary conditions can be adjusted in the `initialize_flow_variables` subroutine, the initial value for temperature is set by modifying the value of `T_initial`. The boundary condition type for each wall group is stored in the `T_Bound_type` integer array as follows:

Value of `T_Bound_type`	**Boundary condition type**
0	Dirichlet Boundary
1	Neuman Boundary
2	Mixed Boundary

The temperature value for each boundary is stored in `T_Bound` array in case of `Dirichlet Boundary`, the flux is stored in the `Flux_bound` array in case of `Neuman Boundary`, and the heat transfer coefficient and T_∞ are stored under `h_infinity` and `T_infinity` for a mixed boundary condition.

The output file is a .vtk file which can be read using the free post processor Paraview.

The steady state diffusion equation:

$$\nabla.(\Gamma.\nabla\phi) = 0$$

Where:

Φ : is a scalar variable (Temperature, Mass fraction, etc.).

Γ : is the Diffusion coefficient.

Integrating the diffusion equation over the control volume:

$$\int \nabla.(\Gamma.\nabla\phi)\, dV = 0$$

Using the convergence theorem the previous volume integral can be converted into an integration of the fluxes over the control volume bounding surfaces as follows:

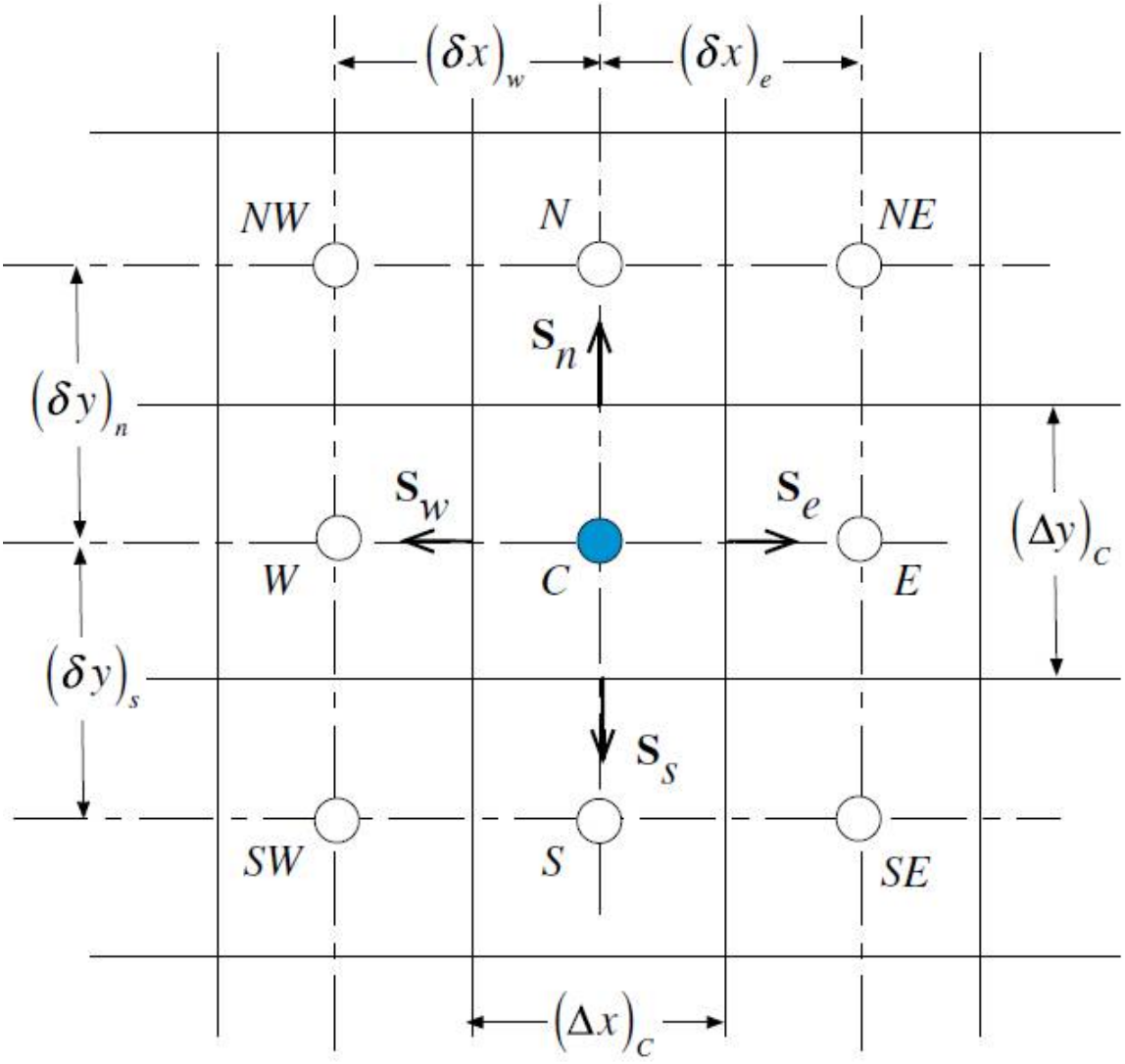
$$\oint (\Gamma.\nabla\phi)\, dA = 0$$

Which can be converted to the summation of the flux through the control volume faces as follows:

$$\sum_{c.v.\ faces} \Gamma.\nabla\phi_{face}.S_{face} = 0$$

Discretization of the diffusion equation:

In case of an orthogonal grid, the cell face will be normal to the line connecting the centroids of neighboring cells as shown in the following figure:



So each term of the previous summation can be discretized as follows (assuming a 2D grid like the one shown in the previous figure and a depth of value 1):

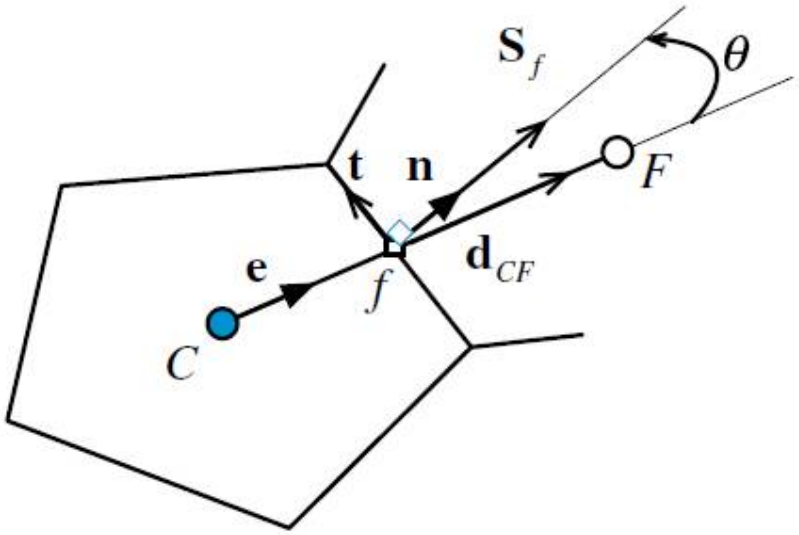
$$\sum_{c.v.\ faces} \Gamma.\nabla\phi_{face}.A_{face} = \Gamma.\frac{\partial\phi_e}{\partial x}.\Delta y + \Gamma.\frac{\partial\phi_w}{\partial x}.\Delta y + \Gamma.\frac{\partial\phi_n}{\partial y}.\Delta x + \Gamma.\frac{\partial\phi_s}{\partial y}.\Delta x$$

And:

$$\Gamma.\frac{\partial\phi_e}{\partial x}.\Delta y = \Gamma.\Delta y.\frac{\phi_E - \phi_C}{\Delta x}$$

The unstructured grid discretization:

In the above configuration, the fluxes were normal to the face. In general, unstructured grids are non-orthogonal. Therefore the surface vector Sf and the vector CF joining the centroids of the elements straddling the interface are not collinear (see the figure below). In this case the gradient normal to the surface cannot be written as a function of ϕ_F and ϕ_C, as it has a component in the direction perpendicular to CF.



Thus to achieve the linearization of the flux in non-orthogonal grids, the surface vector Sf should be written as the sum of two vectors E_f and T_f:

$$S_f = E_f + T_f$$

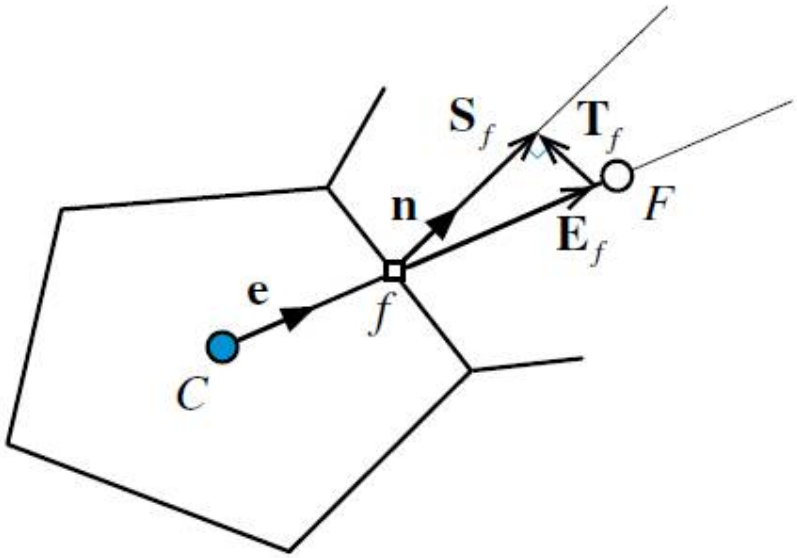
Such that E_f is being selected in the CF direction to enable writing part of the diffusion flux as a function of the values at the cell centers ϕ_F and ϕ_C, so each term in the diffusion flux summation becomes:

$$\nabla\phi_f . S_f = \nabla\phi_f . E_f + \nabla\phi_f . T_f$$

$$\nabla\phi_f . S_f = \frac{\phi_F - \phi_C}{d_{CF}} . \|E_f\| + \nabla\phi_f . T_f$$

The first term in the right hand side represents the contribution similar to that of the orthogonal grid and relates the equation to the neighboring cell center values, the second term is called the cross diffusion term, and is present due to the non-orthogonality of the grid. As mentioned earlier the direction of E_f is being selected in the CF direction, but the length is not determined, there are different methods to select the length of E_f and hence

determine the direction and length of T_f , the one used in this code Is called Over-Relaxed Approach, in this method T_f is selected to be normal to the vector S_f as shown in the following figure.



As a result E_f should be computed as follows:

$$E_f \cos(\theta) = \|S_f\|.e$$

$$E_f = \frac{\|S_f\|}{\cos(\theta)}.e = \frac{\|S_f\|^2}{\|S_f\|.\cos(\theta)}.e = \frac{S_f.S_f}{e.S_f}.e$$

And T_f will be:

$$T_f = S_f - E_f = \|S_f\|(n - \frac{1}{\cos(\theta)}.e)$$

The algebraic equations:

The algebraic equation for each cell will have the general form of:

$$a_C\phi_C + \sum_{nieghbours} a_F\phi_F = b_C$$

Following the previous discretization:

$$a_F = -\Gamma\frac{\|E_f\|}{d_{CF}}$$

$$a_c = \sum_{nieghbours} \Gamma\frac{\|E_f\|}{d_{CF}}$$

$$b_C = \sum_{nieghbours} \Gamma.\nabla\phi_f.T_f$$

Boundary Conditions:

There are three types of boundary conditions implemented in the solver and they are:

Dirichlet boundary Condition:

A Dirichlet boundary condition is when Φ is specified by the user at the boundary, there will still be a need to account for the cross-diffusion, which arises on boundary faces as on interior faces. This happens whenever the surface vector is not collinear with the vector joining the centroids of the element and boundary face. The diffusion flux along the boundary face is discretized as:

$$\nabla\phi_b . S_b = \frac{\phi_b - \phi_C}{d_{Cb}} . \|E_b\| + \nabla\phi_b . T_b$$

The algebraic equation coefficient contribution from this face will become:

$$a_F = 0$$

$$a_c = \Gamma \frac{\|E_b\|}{d_{Cb}}$$

$$b_C = \Gamma \frac{\|E_b\|}{d_{Cb}} + \Gamma . \nabla\phi_b . T_b$$

Neuman boundary Condition:

In this boundary condition type the flux at the boundary face is defined by the user, so the flux value is just injected into the algebraic equations as a source term, in that case the algebraic equation coefficient contribution from this face will become:

$$a_F = 0$$

$$a_c = \Gamma \frac{\|E_b\|}{d_{Cb}}$$

$$b_C = -q_b . S_b$$

And the value at the boundary face will be calculated using the following relation:

$$\phi_b = \frac{\Gamma \frac{\|E_b\|}{d_{Cb}} \phi_C - q_b}{\Gamma \frac{\|E_b\|}{d_{Cb}}}$$

Mixed boundary Condition:

The mixed boundary condition, refers to the situation where information at the boundary is given via a convection transfer coefficient h_∞ and a surrounding value ϕ_∞ as

$$\Gamma . \nabla\phi_b . S_b = \Gamma . \frac{\phi_b - \phi_C}{d_{Cb}} . \|E_b\| + \Gamma . \nabla\phi_b . T_b = h_\infty (\phi_\infty - \phi_b) . S_b$$

In that case the algebraic equation coefficient contribution from this face will become:

$$a_F = 0$$

$$a_c = \frac{h_\infty \|S_b\| \Gamma \frac{\|E_b\|}{d_{Cb}}}{h_\infty \|S_b\| + \Gamma \frac{\|E_b\|}{d_{Cb}}}$$

$$b_C = \frac{h_\infty \|S_b\| \Gamma \frac{\|E_b\|}{d_{Cb}}}{h_\infty \|S_b\| + \Gamma \frac{\|E_b\|}{d_{Cb}}} \phi_\infty + \frac{h_\infty \|S_b\| \Gamma . \nabla \phi_b . T_b}{h_\infty \|S_b\| + \Gamma \frac{\|E_b\|}{d_{Cb}}}$$

And the value at the boundary face will be calculated using the following relation:

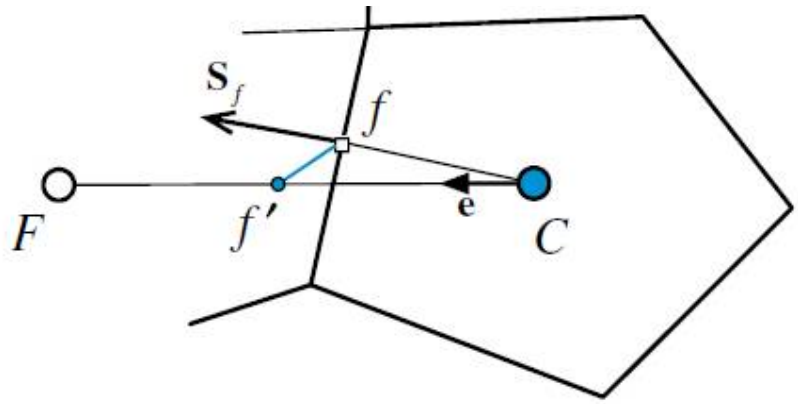
$$\phi_b = \frac{h_\infty \|S_b\| \phi_\infty + \Gamma \frac{\|E_b\|}{d_{Cb}} \phi_C - \Gamma . \nabla \phi_b . T_b}{h_\infty \|S_b\| + \Gamma \frac{\|E_b\|}{d_{Cb}}}$$

Calculating the gradients:

The Green-Gauss gradient method is employed to calculate the gradients at the cell centroids, the gradient is computed as shown in the following equation:

$$\nabla \phi_C = \frac{1}{V_C} \sum_{faces} \phi_f . S_f$$

Where V_C is the cell volume, ϕ_f is the scalar value at the face, and S_f is the face area vector. To proceed with calculation the scalar values at the faces has to be first calculated, this is done by calculating the scalar value at the mid-point between the cell centroids and call that point f' as shown in the figure below.



The scalar value is calculated at f' as follows:

$$\phi_{f'} = \frac{\phi_C + \phi_F}{2}$$

Then the value of the scalar at the face is calculated as follows:

$$\phi_f = \phi_{f'} + \nabla \phi_{f'} . (r_f - r_{f'})$$

Where $r_{f'}$ and r_f are the location coordinates of the f' and the face center f , and $\nabla \phi_{f'}$ can be calculated as:

$$\nabla \phi_{f'} = \frac{\nabla \phi_C + \nabla \phi_F}{2}$$

As shown above to get the face value we need the gradients which we were originally trying to calculate, so the gradient calculation will be an iterative operation as shown in the following steps:

1- Calculate ϕ_f, using $\phi_{f'} = \frac{\phi_C+\phi_F}{2}$
2- Calculate $\nabla\phi_C$ using f' values as an initial guess $\nabla\phi_C = \frac{1}{V_C}\sum_{faces}\phi_{f'}.S_f$.
3- Get the face values using the calculated $\nabla\phi_C$ as follows: $\phi_f = \phi_{f'} + \nabla\phi_{f'}.\left(r_f - r_{f'}\right) = \phi_{f'} + \frac{\nabla\phi_C+\nabla\phi_F}{2}.\left(r_f - \frac{r_C+r_F}{2}\right)$
4- Update $\nabla\phi_C$ using the calculated face values $\nabla\phi_C = \frac{1}{V_C}\sum_{faces}\phi_f.S_f$
5- Repeat from step 3.

Interpolating the gradient to the faces:

It was shown in the discretization of the diffusion term that the non-orthogonal grids requires the use of correction terms involving gradients at control volume faces. Thus in this situation the gradients need to be interpolated from the control volume centroids where they were computed to the control volume faces where they will be used.

By considering the configuration in following figure, which shows the gradients$\nabla\phi_C$ and $\nabla\phi_F$ of the variable ϕ at the two nodes C and F respectively. The interpolated gradient at the face, $\overline{\nabla\phi_f}$ is obtained by averaging the values at nodes C and F, as shown in following figure. It is important for the stencil of the gradient at the face to be heavily based on the nodes straddling the face, which is not guaranteed by this simple averaging practice.

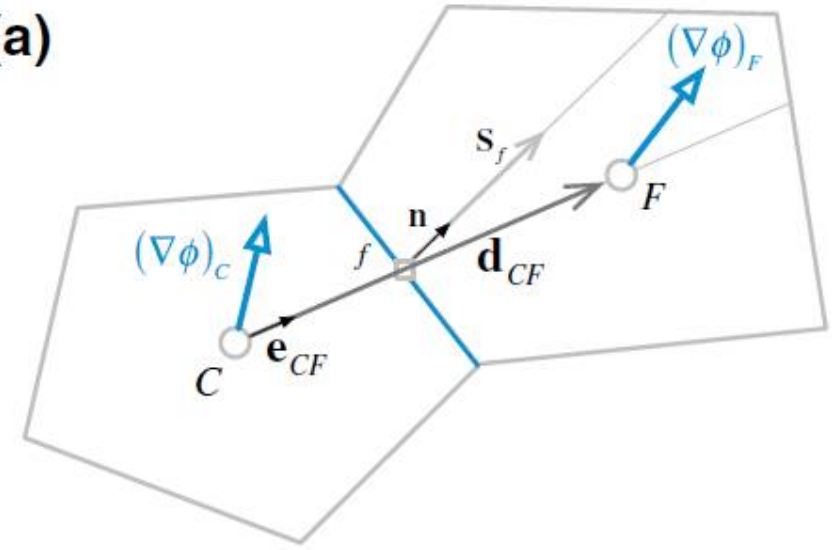
As schematically displayed in following figure, this can be accomplished by forcing the face gradient along the CF direction to be equal to the local gradient defined by the values of ϕ at C and F. Mathematically this can be written as:

$$\nabla\phi_f = \overline{\nabla\phi_f} + \left[\frac{\phi_F - \phi_C}{d_{CF}} - \left(\overline{\nabla\phi_f}.e_{CF}\right)\right]e_{CF}$$

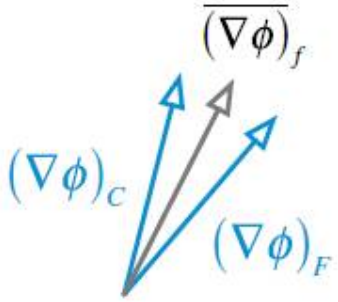
Where:

$$\overline{\nabla\phi_f} = g_c\phi_C + g_F\phi_F\,, \quad g_c = \frac{\|r_f + r_C\|}{\|r_F + r_C\|}\,, \quad e_{CF} = \frac{r_F - r_C}{d_{CF}}$$

(a)



(b)



(c)

$\overline{(\nabla\phi)}_f - \left(\overline{(\nabla\phi)}_f \cdot \mathbf{e}_{CF}\right)\mathbf{e}_{CF}$