SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Finite Volume Discretization Techniques of Differential Operators

Simone Bnà, Cineca

Univerza v Ljubljani

TECHNISCHE UNIVERSITÄT WIEN

CINECA

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

09/2022

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

"This material is based on *An introduction to Computational Fluid Dynamics using OpenFOAM with advanced topics* by Riccardo Rossi, Head and Founder of RED Fluid Dynamics, March/April 2021".

The scalar transport equation

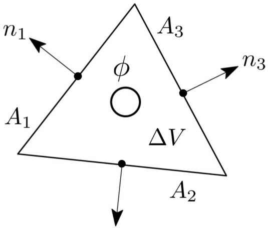
Simone Bnà, Cineca

We use the **finite-volume method** (FVM) to solve the flow governing equations. The **integral form** of the scalar transport equation (STE) must be discretized and solved:

$$\int_V \frac{\partial \phi}{\partial t}\, dV + \int_A (u_j \phi) n_j \, dA = \int_A D \frac{\partial \phi}{\partial x_j} n_j \, dA + \int_V (S_\phi)\, dV$$

Discretization steps:

- Numerical integration
- Time-advancement schemes
- Differentiation schemes
- Interpolation schemes



Properties of numerical schemes

Simone Bnà, Cineca

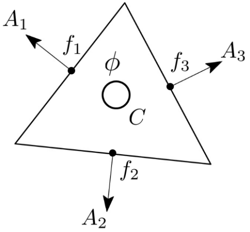
The important properties of a numerical schemes are:

- **Convergence/accuracy**: the numerical solution should converge to the exact solution of the PDE as the mesh size tends to zero
- **Conservation**: underlying conservation laws should be respected at the discrete level
- **Boundedness**: quantities like density, temperature and concentration should remain non-negative and free of spurious wiggles/spikes

And what about the **spectral-resolution**?

If we **integrate in time** the semi-discrete transport equation and we apply the unconditionally stable backward Euler scheme we get:

$$(\phi^{n+1} - \phi^n)\Delta V + \sum_f (u_j^n \phi^{n+1})_f A_{fj} = \sum_f D_f \frac{\partial \phi^{n+1}}{\partial x_j}\Big|_f A_{fj} + S_\phi^{n+1}\Delta V$$

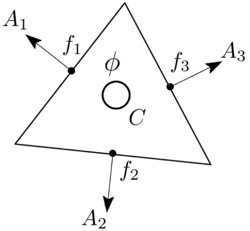


Note that the **midpoint rule** is **second-order accurate** only if variables are evaluated at the **cell/face centroid**.

$$(\phi^{n+1} - \phi^n)\Delta V + \sum_f (u_j^n \phi^{n+1})_f A_{fj} = \sum_f D_f \frac{\partial \phi^{n+1}}{\partial x_j}\Bigg|_f A_{fj} + S_\phi^{n+1} \Delta V$$

We need to define the **discretization method** for the following operators:

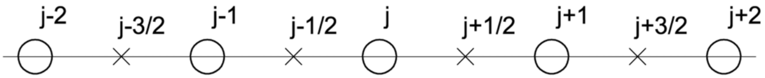
- Cell-centered gradient
- Diffusion operator
- Convection operator



The computation of **cell-centered gradients** is widely used in finite-volume schemes for **convection interpolation** and **diffusion discretization**.

On a uniform, one-dimensional stencil a **second-order accurate** formula based on **Taylor expansions** is the following:

$$\left.\frac{\partial \phi}{\partial x}\right|_j = \frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x} - \frac{1}{6}\left.\frac{\partial^3 \phi}{\partial x^3}\right|_j \Delta x^2 + \mathcal{O}(\Delta x^3)$$



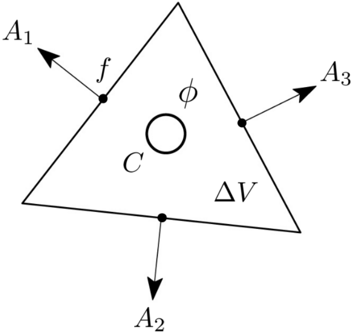
In the framework of **unstructured finite-volume schemes**, cell-centered gradients can be estimated via the **Gauss-Green theorem**:

$$\int_V \frac{\partial \phi}{\partial x_j}\, dV = \int_A \phi n_j\, dA$$

which, using the **midpoint rule** to evaluate the surface and volume integrals, gives:

$$\left.\frac{\partial \phi}{\partial x_j}\right|_c \approx \frac{1}{\Delta V} \sum_f \phi_f A_{fj}$$

where ϕ_f is the face-interpolated value.



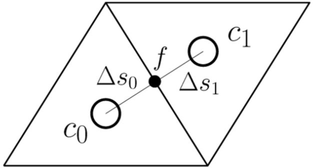
The **face interpolated value** as well can be estimated via **Taylor expansions**:

$$\phi_0 = \phi_f - \left.\frac{\partial \phi}{\partial s}\right|_f \Delta s_0 + \frac{\partial^2 \phi}{\partial s^2}\frac{\Delta s_0^2}{2} + \mathcal{O}(\Delta s_0^3)$$

$$\phi_1 = \phi_f + \left.\frac{\partial \phi}{\partial s}\right|_f \Delta s_1 + \frac{\partial^2 \phi}{\partial s^2}\frac{\Delta s_1^2}{2} + \mathcal{O}(\Delta s_1^3)$$

The **midPoint scheme** provides a simple **second-order** formula on **regular grids** ($\Delta s_0 = \Delta s_1$):

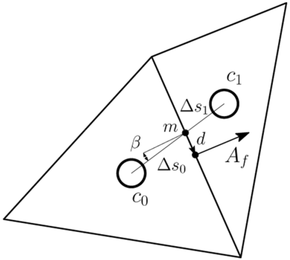
$$\phi_f^{MP} = \frac{\phi_1 + \phi_0}{2} + \mathcal{O}(\Delta s^2)$$



In complex geometries the computational grid usually deviates from regular elements and **quality parameters** are thus introduced.

Mesh quality:

- Aspect ratio ($\Delta s_1 / \Delta s_0$)
- Non-orthogonality (β)
- skewness (d)



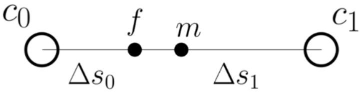
Mid-point rule:

$$\phi_f^M = \frac{\phi_1 + \phi_0}{2} + \mathcal{O}(\Delta s^2)$$

Linear interpolation:

$$\phi_f^L = \phi_0 \lambda_0 + \phi_1 (1 - \lambda_0) + \mathcal{O}(\Delta s^2)$$

$$\lambda_0 = \frac{\Delta s_0}{\Delta s}, \quad \Delta s = \Delta s_0 + \Delta s_1$$



Mid-point rule:

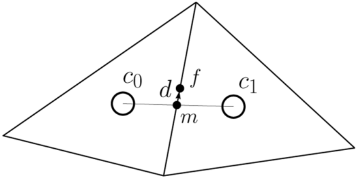
$$\phi_f^M = \frac{\phi_1 + \phi_0}{2} + \mathcal{O}(\Delta s^2)$$

Skew-corrected intepolation:

$$\phi_m^L = \phi_0\lambda_0 + \phi_1\lambda_1\,, \quad \lambda_1 = 1 - \lambda_0$$

$$(\nabla\phi)_m^L = (\nabla\phi)_0\lambda_0 + (\nabla\phi)_1\lambda_1$$

$$\phi_f^{SC} = \phi_m + (\nabla\phi)_m \cdot d$$

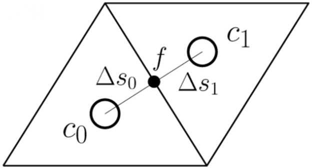


The discretization of the **diffusive flux** in the STE requires to evaluate the **scalar gradient** at the **face centroid**:

$$\int_A D\frac{\partial\phi}{\partial x_j}n_j \approx \sum_f D\frac{\partial\phi}{\partial x_j}\bigg|_f A_{fj}$$

On **regular grids**, the face gradient can be evaluated via the **central scheme**:

$$\frac{\partial\phi}{\partial x_j} \approx \frac{\phi_1 - \phi_0}{\Delta s} + \mathcal{O}(\Delta s^2)$$



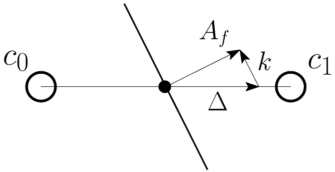
On **irregular** meshes, a **correction** for the **grid non-orthogonality**, when severe, might be needed to improve the accuracy of the numerical solution:

$$\left.\frac{\partial \phi}{\partial x_j}\right|_f A_{fj} = |\Delta| \frac{\phi_1 - \phi_0}{\Delta s} + k \cdot (\nabla \phi)_f$$

where

$$(\nabla \phi)_f = (\nabla \phi)_0 \lambda_0 + (\nabla \phi)_1 \lambda_1$$

is used to build up the **correction term**.



Convection interpolation operators I

Simone Bnà, Cineca

The discretization of the **convection operator** requires to evaluate the field at the **face centroid**.

A second-order accurate **central interpolation** on irregular meshes can be obtained using a **two-sided linear expansion** [1]:

$$\phi_f = \frac{1}{2}(\phi_0 + \phi_1) + \frac{1}{2}\left[\left.\frac{\partial \phi}{\partial x_i}\right|_0 R_{0i} + \left.\frac{\partial \phi}{\partial x_i}\right|_1 R_{1i} \right] + \mathcal{O}(\Delta s^2)$$

Alternatively, using a **one-sided linear expansion**, **upwind schemes** are obtained:

$$\phi_f = \phi_u + \left.\frac{\partial \phi}{\partial x_i}\right|_u R_{ui} + \mathcal{O}(\Delta s^2)$$

[1] P. Batten, C. Lambert, D.M. Causon, Positively conservative high-resolution convection schemes for unstructured elements, Int. J. Numer. Methods Eng. 39 (1996) 1821.

Convection interpolation operators II

Simone Bnà, Cineca

Several convection interpolation schemes have been proposed in the context of the FV framework. We report here the **low-order** and **high-order** upwind and central interpolation schemes:

Upwind (FOU) : $\phi_f = \phi_u + \mathcal{O}(\Delta s)$

linearUpwind (SOU) : $\phi_f = \phi_u + \left.\frac{\partial \phi}{\partial x_i}\right|_u R_{ui} + \mathcal{O}(\Delta s^2)$

midPoint (LOCD) : $\phi_f = \frac{1}{2}(\phi_0 + \phi_1) + \mathcal{O}(\Delta s^2)$

reconCentral (HOCD) : $\phi_f = \frac{1}{2}(\phi_0 + \phi_1) + \frac{1}{2}\left[\left.\frac{\partial \phi}{\partial x_i}\right|_0 R_{0i} + \left.\frac{\partial \phi}{\partial x_i}\right|_1 R_{1i}\right] + \mathcal{O}(\Delta s^2)$

Slope limiter

Simone Bnà, Cineca

In the framework of unstructured finite-volume schemes, **slope-limiting techniques** can be used to keep the **face-interpolation bounded**.

$$\phi_f = \phi_u + min(\alpha_f) \left.\frac{\partial \phi}{\partial x_j}\right|_u R_{uj}$$

For each cell, the **limiter** is given by:

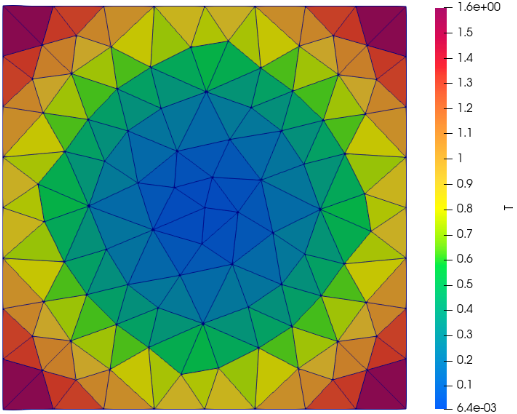
$$\alpha_f = \begin{cases} min\left(1, \frac{\phi_{max} - \phi_u}{\phi_f - \phi_u}\right) & \text{if}(\phi_f - \phi_u) > 0 \\ min\left(1, \frac{\phi_{min} - \phi_u}{\phi_f - \phi_u}\right) & \text{if}(\phi_f - \phi_u) < 0 \\ 1 & \text{otherwise} \end{cases}$$

where ϕ_u is the cell-centroid value at the upwind side of the face.

In OpenFOAM, the schemes for **gradient** (cell-centered) computation and **diffusion/convection** discretization are specified in the **fvSchemes** dictionary file located in the **system** folder:

```
gradSchemes
{
    default linear;
    grad(u) cellLimited Gauss skewCorrected midPoint 1;
}
divSchemes
{
    div(phi, u) Gauss linearUpwind grad(u);
}
laplacianSchemes
{
    Laplacian(D, u) Gauss linear corrected;
}
```

Simone Bnà, Cineca



Test function:

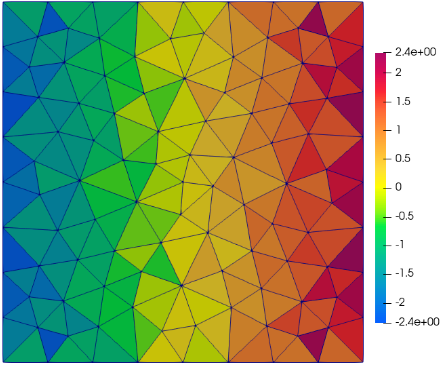
$$T(x, y) = x^2 + y^2$$

$$\frac{\partial T}{\partial x} = 2x, \quad \frac{\partial T}{\partial y} = 2y$$

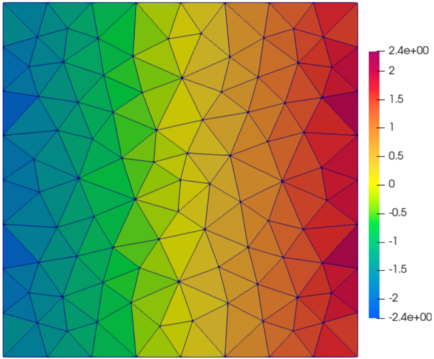
example I: computation of cell gradient

Simone Bnà, Cineca

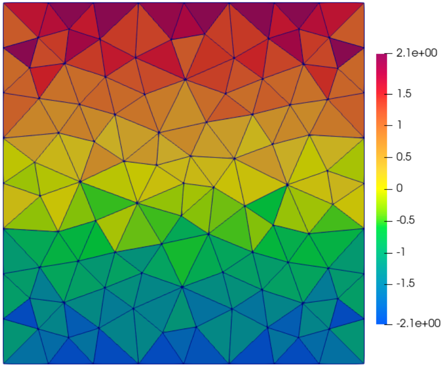
SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP



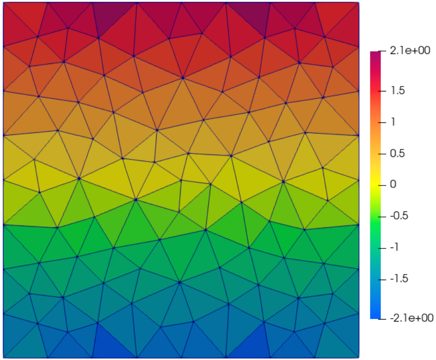
Sx - Gauss midPoint



Sx - Gauss skewCorrected linear

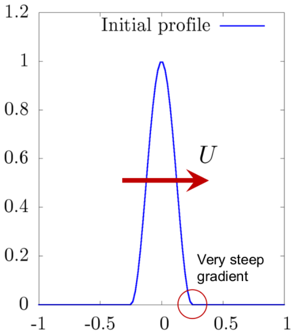


Sy - Gauss midPoint



Sy - Gauss skewCorrected linear

The (pure) **advection** of a scalar profile in uniform flow can be used to investigate the **conservation and spectral properties** of CIOs.



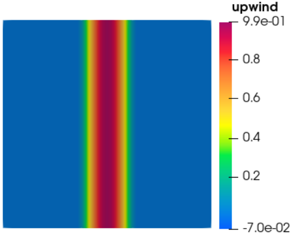
Flow parameters:

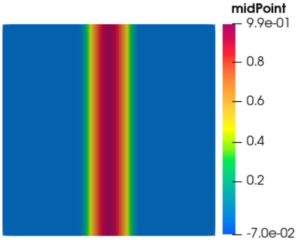
$$\frac{\partial \phi}{\partial t} + U\frac{\partial \phi}{\partial x} = 0\,,\, U = 1$$

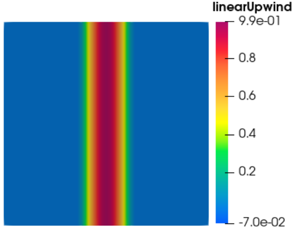
$$\phi(x,0) = \begin{cases} cos^2(2\pi x) & \text{if} |x| \leq 0.25 \\ 0 & \text{if} |x| > 0.25 \end{cases}$$

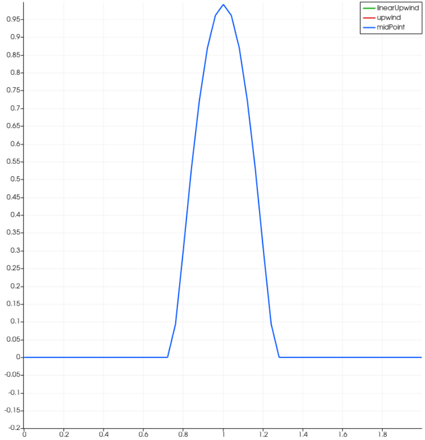
example II: Advection of a scalar profile II

Simone Bnà, Cineca



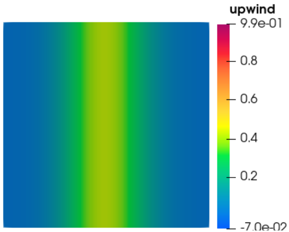


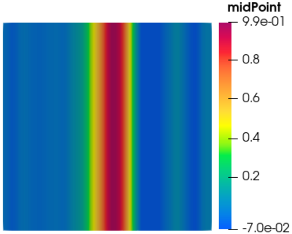


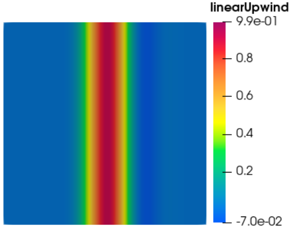


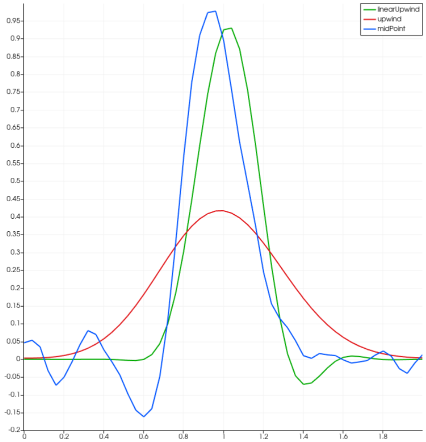
example II: Advection of a scalar profile II

Simone Bnà, Cineca









Thank you for your attention!