Differential operators on polyhedral unstructured meshes

Goal

Let us consider a scalar function $\phi : \mathbb{R}^3 \to \mathbb{R}$, and a mesh composed by N polyhedral cells $\{c_i\}_{1,\ldots,N}$. We assume to know the value of ϕ on each cell. We denote by ϕ_i the value of ϕ on the i^{th} cell. Similarly, we denote by $(\partial_x \phi)_i$ the value of the x derivative of ϕ on the i^{th} cell. The goal is to construct a matrix $D^{(x)}$ which calculates the x derivative:

$$(\partial_x \phi)_i = \sum_j D^{(x)}_{ij} \phi_j \tag{1}$$

and similarly for the y and z derivatives.

Concept

The method is based on the Green-Gauss theorem (see Sec. III.A of Ref. [1]). The average gradient of ϕ over one cell is:

$$(\overline{\boldsymbol{\nabla}\phi})_i = \frac{1}{V_i} \int_{c_i} dV \, \boldsymbol{\nabla}\phi = \frac{1}{V_i} \oint_{\partial c_i} dA \, \phi \, \hat{\boldsymbol{n}} \tag{2}$$

where V_i denotes the volume of a cell, ∂c_i the external surface, and $\hat{\boldsymbol{n}}$ the unit vector normal to the surface at a given point. For a polyhedral cell we have:

$$(\overline{\boldsymbol{\nabla}\phi})_i \to \frac{1}{V_i} \sum_f \overline{\phi}_f \hat{\boldsymbol{n}}_f A_f \tag{3}$$

where the index f runs over the external faces of the i^{th} cell, $\hat{\boldsymbol{n}}_f$ denotes the unit vector normal to the f^{th} face, A_f its area, and $\overline{\phi}_f$ is the average value of ϕ over the face. It is thus necessary to evaluate the face average $\overline{\phi}_f$. As pointed out in Ref. [1], several alternative methods exist, the simplest being averaging the two values assumed by ϕ over the the two cells i and i' that share the face f (see Sec. III.A.1 of Ref. [1]).

Implementation

The set of all the normal vectors to the faces is denoted by:

$$\{\hat{\boldsymbol{n}}_f\} \tag{4}$$

We introduce the matrix B having number of rows equal to the total number of faces, and number of columns equal to the total number of cells. The entry B_{fj} is equal to $+1$ if the face f belongs to the cell j and the normal is direct outwards, is equal to -1 if the face f belongs to the cell j and the normal is direct inwards, and is equal to 0 otherwise (i.e. if the face f does not belong to the cell j). By letting the index f run over *all* the faces (instead of only those belonging to the i^{th} cell), we can modify Eq. 3:

$$(\overline{\boldsymbol{\nabla}\phi})_i = \frac{1}{V_i} \sum_f B_{fi} \overline{\phi}_f \hat{\boldsymbol{n}}_f A_f \tag{5}$$

As mentioned, we can calculate $\overline{\phi}_f$ by averaging the value of ϕ over the two cells adjacent to the face f:

$$\overline{\phi}_f = \frac{1}{\sum_h |B_{fh}|} \sum_j |B_{fj}| \phi_j \tag{6}$$

The quantity $\sum_h |B_{fh}|$ corresponds to the number of cells sharing the face f, (i.e. 2 is the face is shared between two cells, and 1 if the face is on the boundary). Plugging Eq. 6 into Eq. 5 we get:

$$(\overline{\boldsymbol{\nabla}\phi})_i = \frac{1}{V_i} \sum_f B_{fi} \hat{\boldsymbol{n}}_f A_f \frac{1}{\sum_h |B_{fh}|} \sum_j |B_{fj}| \phi_j = \sum_j \boldsymbol{D}_{ij} \phi_j \tag{7}$$

with

$$\boldsymbol{D}_{ij} = \frac{1}{V_i} \sum_f B_{fi} \hat{\boldsymbol{n}}_f A_f \frac{1}{\sum_h |B_{fh}|} |B_{fj}| \tag{8}$$

The matrix $D^{(x)}$ is obtained from $\boldsymbol{D}$ by taking the scalar product with the unit vector $\hat{\boldsymbol{e}}_x$, and similarly for the y and z components of the gradient.

Least Squares Interpolation

It can be shown (Sec. IV.B of Ref. [1]) that the naïve averaging approach discussed above carries a zero'th order error term for non-uniform cell spacing, that is, though the estimated face value $\overline{\phi}_f$ converges as the cell size becomes infinitely small, it does not converge to the correct value. A slightly more sophisticated method follows from Taylor expanding the value ϕ_i at the i^{th} cell around the center of the face,

$$\phi_i = \phi_f + \Delta x_i \phi_{f,x} + \Delta y_i \phi_{f,y} + \Delta z_i \phi_{f,z} + \dots, \tag{9}$$

where ... refers to higher order terms. If we introduce the functional

$$F(w_i, \phi_i, \phi_{f,x}, \phi_{f,y}, \phi_{f,z}, \dots) = \sum_{i=1}^{N} w_i \left[\Delta\phi_i + \Delta x_i \phi_{f,x} + \Delta y_i \phi_{f,y} + \Delta z_i \phi_{f,z} + \dots\right]^2, \tag{10}$$

where w_i are user-defined weights for each point i and $\Delta\phi_i = \phi_f - \phi_i$, then all of the following is standard procedure for the least squares method. Minimizing F yields the equations

$$\frac{\partial F}{\partial \phi_f} = \sum_{i=1}^{N} 2w_i \left[\Delta\phi_i + \Delta x_i \phi_{f,x} + \Delta y_i \phi_{f,y} + \Delta z_i \phi_{f,z} + \dots\right] = 0 \tag{11}$$

$$\frac{\partial F}{\partial \phi_{f,x}} = \sum_{i=1}^{N} 2w_i \Delta x \left[\Delta\phi_i + \Delta x_i \phi_{f,x} + \Delta y_i \phi_{f,y} + \Delta z_i \phi_{f,z} + \dots\right] = 0 \tag{12}$$

$$\frac{\partial F}{\partial \phi_{f,y}} = \sum_{i=1}^{N} 2w_i \Delta y \left[\Delta\phi_i + \Delta x_i \phi_{f,x} + \Delta y_i \phi_{f,y} + \Delta z_i \phi_{f,z} + \dots\right] = 0 \tag{13}$$

$$\frac{\partial F}{\partial \phi_{f,z}} = \sum_{i=1}^{N} 2w_i \Delta z \left[\Delta\phi_i + \Delta x_i \phi_{f,x} + \Delta y_i \phi_{f,y} + \Delta z_i \phi_{f,z} + \dots\right] = 0 \tag{14}$$

which allow us to write up a linear system of equations,

$$G\Phi_f = H\phi \to \Phi_f = G^{-1}H\phi, \tag{15}$$

where $\Phi_f = [\phi_f, \phi_{f,x}, \phi_{f,y}, \phi_{f,z}]^T$, $\phi = [\phi_1, \phi_2, \dots, \phi_i, \dots, \phi_N]$,

$$G = \begin{bmatrix} \sum w_i & \sum w_i\Delta x_i & \sum w_i\Delta y_i & \sum w_i\Delta z_i \\ \sum w_i\Delta x_i & \sum w_i\Delta x_i^2 & \sum w_i\Delta x_i\Delta y_i & \sum w_i\Delta x_i\Delta z_i \\ \sum w_i\Delta y_i & \sum w_i\Delta x_i\Delta y_i & \sum w_i\Delta y_i^2 & \sum w_i\Delta y_i\Delta z_i \\ \sum w_i\Delta z_i & \sum w_i\Delta x_i\Delta z_i & \sum w_i\Delta y_i\Delta z_i & \sum w_i\Delta z_i^2 \end{bmatrix} \text{ and } H = \begin{bmatrix} w_1 & w_2 & \dots \\ w_1\Delta x_i & w_2\Delta x_i & \dots \\ w_1\Delta y_i & w_2\Delta y_i & \dots \\ w_1\Delta z_i & w_2\Delta z_i & \dots \end{bmatrix}. \tag{16}$$

Since we are interested in ϕ_f, only the first row of $G^{-1}H$ is of interest, saving computational effort and leaving the final estimate

$$\overline{\phi}_f = \sum_{i=1}^{N}\sum_{k=1}^{o} G^{-1}_{1,k} H_{k,i}\phi_i, \tag{17}$$

with o given by the desired accuracy of the Taylor approximation (here $o = 4$ for the three-dimensional 1^{st} order expansion). The weights w_i can be chosen freely, usually as a function of geometrical parameters such as inverse distance or cell volumes (Ref. [2]). In Ref. [1], this method is tested for different 2D meshes with weights $w_i = 1/d_{if}$ as the inverse distance between the face center and the center of cell i. Results compare favorably with other unstructured mesh gradient methods and are at least 1^{st} order accurate excluding, possibly, numerical rounding errors.

Calculating the Laplacian

While the above yields an estimate of the gradient $(\overline{\boldsymbol{\nabla}\phi})_i$ it can easily be extended to an estimated Laplacian using not the Green-Gauss theorem but the divergence theorem:

$$(\overline{\boldsymbol{\nabla}^2\phi})_i = \frac{1}{V_i}\int_{c_i} dV\,\boldsymbol{\nabla}^2\phi = \frac{1}{V_i}\oint_{\partial c_i} dA\,\boldsymbol{\nabla}\phi\cdot\hat{\boldsymbol{n}}, \tag{18}$$

leading to

$$(\overline{\boldsymbol{\nabla}^2\phi})_i \to \frac{1}{V_i}\sum_f (\overline{\boldsymbol{\nabla}\phi})_f \cdot \hat{\boldsymbol{n}}_f A_f \tag{19}$$

for polyhedral cells. Naturally, this requires an estimate of the gradient at the faces, $(\overline{\boldsymbol{\nabla}\phi})_f$, but returning to the linear system of equations of Eq. 15, we find

$$\overline{\phi}_{f,x} = \sum_{i=1}^{N}\sum_{k=1}^{o} G^{-1}_{2,k}H_{k,i}\phi_i, \qquad \overline{\phi}_{f,y} = \sum_{i=1}^{N}\sum_{k=1}^{o} G^{-1}_{3,k}H_{k,i}\phi_i \quad\text{and}\quad \overline{\phi}_{f,z} = \sum_{i=1}^{N}\sum_{k=1}^{o} G^{-1}_{4,k}H_{k,i}\phi_i. \tag{20}$$

It is at present unclear if this method has been tested in the literature. The above can of course be extended to higher order Taylor approximations and alternative weighting functions. Since G^{-1} is only calculated at the beginning of the procedure, it would very inexpensive to have different weightings $w_{x,i}$, $w_{y,i}$ and $w_{z,i}$ for the different components of the gradient, but as a first foray, 1^{st} order Taylor approximations with inverse distance weights should do nicely.

Voronoi diagram automatic strategy

.

References

[1] E. Sozer, C. Brehm, and C.C. Kiris, *Gradient Calculation Methods on Arbitrary Polyhedral Unstructured Meshes for Cell-Centered CFD Solvers*, 52nd Aerospace Sciences Meeting (2014).

[2] D. Pérez-Grande, O. Gonzalez-Martinez, P. Fajardo, and E. Ahedo, *Analysis of the Numerical Diffusion in Anisotropic Mediums: Benchmarks for Magnetic Field Aligned Meshes in Space Propulsion Simulations*, Applied Sciences 6 (2016).