Specialized Numerical Differential Operators in the Finite Pointset Method (FPM)

Jörg Kuhnert*

* *Fraunhofer Institut für Techno- und Wirtschaftsmathematik*
Fraunhofer Platz 1
67663 Kaiserslautern, Germany
e-mail: joerg.kuhnert@itwm.fraunhofer.de, web page: http://www.itwm.fhg.de

Abstract

The Finite Pointset Method (FPM) is a meshfree approach to numerically solve partial differential equations (PDE). The method uses a cloud of numerical points, each of which carrying necessary numerical data. In order to solve a PDE, FPM needs to establish approximations of derivatives, which are provided by a specialized least squares algorithm. The least squares idea turns the whole method into a generalized finite difference method. The main field of application of FPM has been computational fluid dynamics (CFD), with particularly applications in gasdynamics as well as incompressible flows. The purpose of the present article is to show the approximation of the differential operators, used by FPM.

1. Definition of Differential operators

Let us suppose there is a pointcloud being sufficiently dense. The positions of the particles are given by

$$\mathbf{x}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \quad i = 1...N \tag{1.1}$$

Where N is the number of particles in the pointcloud. Suppose furthermore that any function u is given only at the discrete particle locations, i.e.

$$u_i \equiv u\left(\mathbf{x}_i\right) \tag{1.2}$$

The vector of discrete function values is given by

$$\mathbf{u} \equiv \left(u_1, u_2, ..., u_N\right)^T \tag{1.3}$$

We call the numerical differential operators in FPM those vectors which provide an approximation of some derivative in the sense

$$\partial^*_{num} u\left(\mathbf{x}_i\right) = \tilde{\partial}^* u\left(\mathbf{x}_i\right) = \tilde{\partial} u_i = \sum_{j=1}^{N} c^*_{ij} \cdot u_j = \left(\mathbf{c}^*_i\right)^T \cdot \mathbf{u} \tag{1.4}$$

The star (*) in the equation above is a placeholder for all the numerous differential operators needed by FPM. In this nomenclature, we have

$$\begin{aligned} \mathbf{c}_i^0 &= \text{ the numerical operator for function approximation} \\ \mathbf{c}_i^x &= \text{ the numerical operator for the x-derivative} \\ \mathbf{c}_i^y &= \text{ the numerical operator for the y-derivative} \\ \mathbf{c}_i^z &= \text{ the numerical operator for the z-derivative} \\ \mathbf{c}_i^\Delta &= \text{ the numerical operator for the Laplacien} \end{aligned} \tag{1.5}$$

just to name the most often appearing ones. Our intension is to develop operators which are, as shown in (1.4), independent on the underlying function values. Having operators which work generally for all given functions will save a lot of computation time.

We also introduce a weight function which switches on the particular neighbors close to some particle i.

$$W_{ij} = w\left(r\left(\mathbf{x}_i, \mathbf{x}_j\right)\right) = w\left(r_{ij}\right) \tag{1.6}$$

The distance function $r\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is given by

$$r\left(\mathbf{x}_i, \mathbf{x}_j\right) = 2\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|}{\left(h\left(\mathbf{x}_i\right) + h\left(\mathbf{x}_j\right)\right)} \tag{1.7}$$

The function $h\left(\mathbf{x}_i\right)$ is referred to as smoothing length, or more commonly interaction radius. It actually rules the local density of the particles, i.e. the mean distance between particles. We try to provide a weight function which becomes zero if $r = 1$ and one if $r = 0$ and which is n-times continuously differentiable. Some examples:

$$w(r) = \begin{cases} \exp\left(-\gamma \cdot r^2\right) - \exp(-\gamma), & \text{if } r < 1 \\ 0, & \text{otherwise} \end{cases} \tag{1.8}$$

$$w(r) = \begin{cases} \left(1 - r^2\right)^{\gamma}, & \text{if } r < 1 \\ 0, & \text{otherwise} \end{cases} \tag{1.9}$$

The second example has the advantage, that its values as well as its first derivative are continuous at the location $r = 1$. The first example is continuous only in the function values, the first derivative slightly jumps. We can group the discrete weight values in the so called weight matrix

$$\mathbf{W}_i = \begin{pmatrix} w_{i1} & & & 0 \\ & w_{i2} & & \\ & & \ddots & \\ 0 & & & w_{iN} \end{pmatrix} \tag{1.10}$$

having the discrete weights on its diagonal and zeros otherwise.

2. General least squares procedure for operators (Versions 1...4)

2.1. Test functions and least squares procedure

We are searching for the operator $\mathbf{c}_i$ fulfilling the least squares criterion

$$\frac{1}{2}\left\|\mathbf{W}_i^{-1}\cdot\mathbf{c}_i\right\|^2=\frac{1}{2}\mathbf{c}_i^T\cdot\mathbf{W}_i^{-1T}\cdot\mathbf{W}_i^{-1}\cdot\mathbf{c}_i\overset{!}{=}\frac{1}{2}\mathbf{c}_i^T\cdot\mathbf{W}_i^{-2}\cdot\mathbf{c}_i\ \min \tag{2.1}$$

under certain consistency conditions

$$\mathbf{K}_i^T\cdot\mathbf{c}_i=\mathbf{b} \tag{2.2}$$

The matrix $\mathbf{K}_i$ represents test functions (given as discrete values for each particle) for which the numerical operator shall give a distinct value. As for example, the numerical operator for the x-derivative $\mathbf{c}_i^x$ shall deliver 0.0 if operating on a constant function $\mathbf{k}_i^0=(1,\ 1,...,1)$ or a quadratic function $\mathbf{k}_i^2=\left(\left(x_1-x_i\right)^2,\ \left(x_2-x_i\right)^2,...,\left(x_N-x_i\right)^2\right)$, but it shall deliver 1.0 if operating on the linear function $\mathbf{k}_i^1=\left(x_1-x_i,\ x_2-x_i,...,x_N-x_i\right)$. In other words, we have the conditions (among others!)

$$\begin{aligned}\left(\mathbf{k}_i^0\right)^T\cdot\mathbf{c}_i^x&=0\\ \left(\mathbf{k}_i^1\right)^T\cdot\mathbf{c}_i^x&=1\\ \left(\mathbf{k}_i^2\right)^T\cdot\mathbf{c}_i^x&=0\end{aligned} \tag{2.3}$$

In general, the matrix $\mathbf{K}$ contains M discretely given test functions. The right hand side vector $\mathbf{b}$ consequently contains the corresponding values to be delivered by the operator if applied to exactly these test functions.
The minimization problem (2.1) together with the constraints (2.2) can be solved (uniquely!) using Lagrangian multipliers in the sense

$$\mathbf{W}_i^{-2}\cdot\mathbf{c}_i+\sum_{k=1}^{M}\lambda_k\cdot\mathbf{k}_i^k=0 \tag{2.4}$$

Let us transform equation (2.4) into

$$\mathbf{c}_i + \sum_{k=1}^{M} \lambda_k \cdot \mathbf{W}_i^2 \cdot \mathbf{k}_i^k = 0 \tag{2.5}$$

And multiply from left with the p-th condition vector $\mathbf{k}_i^p$

$$\left(\mathbf{k}_i^p\right)^T \cdot \mathbf{c}_i + \sum_{k=1}^{M} \lambda_k \left(\mathbf{k}_i^p\right)^T \cdot \mathbf{W}_i^2 \cdot \mathbf{k}_i^k = 0 \tag{2.6}$$

Equation (2.5) can be written more compactly as

$$\mathbf{c}_i + \mathbf{W}_i^2 \cdot \mathbf{K}_i \cdot \boldsymbol{\lambda}_i = 0 \tag{2.7}$$

where $\boldsymbol{\lambda}_i$ is the vector of all M values of lambda.

Equation (2.7) immediately shows that the operator to be searched for is a linear combination out of the test functions together with the weight potential.

(2.6) more compactly reads

$$\mathbf{K}_i^T \cdot \mathbf{c}_i + \mathbf{K}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \mathbf{K}_i \cdot \boldsymbol{\lambda}_i = 0 \tag{2.8}$$

which, by virtue of (2.2), provides

$$\mathbf{b}_i + \mathbf{K}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \mathbf{K}_i \cdot \boldsymbol{\lambda}_i = 0 \tag{2.9}$$

Equation (2.9) provides already a $M \times M$ linear system to be solved for $\cdot\boldsymbol{\lambda}_i$ which then serves for providing the solution

$$\mathbf{c}_i = -\mathbf{W}_i^2 \cdot \mathbf{K}_i \cdot \boldsymbol{\lambda}_i \tag{2.10}$$

The computationally most expensive part here is the production of the matrix $\mathbf{K}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \mathbf{K}_i$ together with the solution of the linear system (2.8). This embeds one risk which lies in the fact that we produce the square of the matrix $\mathbf{W}_i \cdot \mathbf{K}_i$. This matrix is usually well conditioned. Occasionally, its conditioning becomes bad, usually when the local point cloud becomes distorted. Then, the square operation $\mathbf{K}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \mathbf{K}_i$ even makes the conditioning worse, possibly leading to a square matrix which is close to singularity.

2.2. Order of approximation

The order of approximation is given by the choice of the test functions.

Order 1: If only the constant test function is used $\left(\mathbf{k}_i^0\right)^T = (1,1,...,1)$, then constant functions are approximated correctly, so the truncation errors are of linear order.

Order 2: If constant and linear test functions are used $\left(\mathbf{k}_i^0\right)^T = (1,1,...,1)$, $\mathbf{k}_i^1 = \left(x_1 - x_i\,,\; x_2 - x_i\,,...,x_N - x_i\right)$, $\mathbf{k}_i^2 = \left(y_1 - y_i\,,\; y_2 - y_i\,,...,y_N - y_i\right)$, $\mathbf{k}_i^3 = \left(z_1 - z_i\,,\; z_2 - z_i\,,...,z_N - z_i\right)$ then constant and linear functions are approximated correctly, so the truncation errors are of quadratic order.

Order 3: requires also quadratic test functions.

Order n: requires all monomials up to order n-1 in the set of test functions.

3. Better numerical conditioning of the local stencil (Version 5)

Let us return to the consistency conditions (2.2). For the solution of $\mathbf{c}_i$, we would like to avoid the matrix multiplication (2.9) in order to maintain the conditioning of the arising linear system. We start with (2.2)

$$\mathbf{K}_i^T \cdot \mathbf{c}_i = \mathbf{b} \tag{3.1}$$

By multiplication of the identity $\mathbf{I} = \mathbf{W}_i \cdot \mathbf{W}_i^{-1}$, we obtain

$$\begin{aligned} &\mathbf{K}_i^T \cdot \mathbf{c}_i = \\ &\mathbf{K}_i^T \cdot \mathbf{I} \cdot \mathbf{c}_i = \\ &\mathbf{K}_i^T \cdot \mathbf{W}_i \cdot \mathbf{W}_i^{-1} \cdot \mathbf{c}_i = \mathbf{b} \end{aligned} \tag{3.2}$$

Let us define the vector

$$\mathbf{q}_i \equiv \mathbf{W}_i^{-1} \cdot \mathbf{c}_i \tag{3.3}$$

With this definition, (3.2) appears as

$$\mathbf{K}_i^T \cdot \mathbf{W}_i \cdot \mathbf{q}_i = \mathbf{b} \tag{3.4}$$

The linear system (3.4) is underdetermined, as we have M test functions and N neighbor particles. Usually, we have $M < N$, at least we can choose

the number of neighbors such that is exceeds the number of test functions. Similarly to (2.1), we require the solution of (3.4) under the minimization constraint

$$\left\| \mathbf{q}_i \right\|^2 = \min \tag{3.5}$$

The minimization can easily be provided by using the Q-R-decomposition of the $N \times M$-matrix $\mathbf{K}_i^T \cdot \mathbf{W}_i$, i.e.

$$\mathbf{W}_i \cdot \mathbf{K}_i = \mathbf{Q}_{\mathbf{WK}} \cdot \mathbf{R}_{\mathbf{WK}} \tag{3.6}$$

Leading to the linear system

$$\mathbf{R}_{\mathbf{WK}}^T \cdot \mathbf{Q}_{\mathbf{WK}}^T \cdot \mathbf{q}_i = \mathbf{b} \tag{3.7}$$

In fact, as $\mathbf{Q}_{\mathbf{WK}} \cdot$ is a unitary transformation, we search the minimal norm solution of the vector $\mathbf{s}_i \equiv \mathbf{Q}_{\mathbf{WK}}^T \cdot \mathbf{q}_i$ in the linear system

$$\mathbf{R}_{\mathbf{WK}}^T \cdot \mathbf{s}_i = \mathbf{b} \tag{3.8}$$

The minimal solution of $\mathbf{s}_i$ is given by

$$\mathbf{s}_i = \left(\mathbf{R}_{\mathbf{WK}}^T \right)^{-1} \cdot \mathbf{b} \tag{3.9}$$

Here, by $\left(\mathbf{R}_{\mathbf{WK}}^T \right)^{-1}$ we denote the "quasi inverse" in the following sense. $\left(\mathbf{R}_{\mathbf{WK}}^T \right)^{-1}$ has the structure

$$\mathbf{R}_{\mathbf{WK}}^T = \begin{pmatrix} R_{11} & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & 0 & \cdots & 0 \\ R_{1M} & \cdots & R_{MM} & 0 & \cdots & 0 \end{pmatrix} = \left(\hat{\mathbf{R}}_{\mathbf{WK}}^T \quad \mathbf{0} \right) \tag{3.10}$$

We define the quasi inverse by

$$\left(\mathbf{R}_{\mathbf{WK}}^T \right)^{-1} = \begin{pmatrix} \left(\hat{\mathbf{R}}_{\mathbf{WK}}^T \right)^{-1} \\ \mathbf{0} \end{pmatrix} \tag{3.11}$$

The final solution for the differential operator is easily computed by

$$\mathbf{c}_i = \mathbf{W}_i \cdot \mathbf{q}_i = \mathbf{W}_i \cdot \mathbf{Q}_{\mathbf{KW}} \cdot \mathbf{s}_i = \mathbf{W}_i \cdot \mathbf{Q}_{\mathbf{KW}} \cdot \left(\mathbf{R}_{\mathbf{WK}}^T\right)^{-1} \cdot \mathbf{b} \tag{3.12}$$

This procedure helps to keep numerical round-offs small.

4. Using Pseudo-inverse for distorted local stencils (Version 6)

The procedure (3.12) is numerically well conditioned and helps to avoid numerical round-offs if local particle stencils become distorted. It will not help avoiding unstable differential operators, if the pointcloud locally deforms to a less dimensional manifold, i.e. if (for a 3D computation!) the particles are nearly placed in a plane or on a line etc. In this case, only pseudo-inverse considerations of the system (2.9) can help.

Let us again consider this system. We are interested in the solution $\boldsymbol{\lambda}_i$, as this is needed for the assembly of the differential operator according to equation (2.10).

Let us define the matrix

$$\boldsymbol{\Psi}_i \equiv \mathbf{K}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \mathbf{K}_i \tag{4.1}$$

We have to decompose $\boldsymbol{\Psi}_i$ into diagonal form, i.e.

$$\boldsymbol{\Psi}_i = \mathbf{L}_i^T \cdot \boldsymbol{\Sigma}_i \cdot \mathbf{L}_i \tag{4.2}$$

The matrix $\mathbf{L}_i$ contains the mutually orthogonal (left) eigenvectors of the symmetric matrix $\boldsymbol{\Psi}_i$, and $\boldsymbol{\Sigma}_i$ is diagonal and contains the (real) eigenvalues. If the local particle stencil becomes reduced in some dimension, or if not enough neighbor particles are present for a proper (non-singular) setup of the matrix $\boldsymbol{\Psi}_i$, then some of the eigenvalues in $\boldsymbol{\Sigma}_i$ will become very small or even zero. With (4.2), the system (2.9) now can be written as

$$\begin{aligned} &\mathbf{b} + \mathbf{K}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \mathbf{K}_i \cdot \boldsymbol{\lambda}_i = 0 \quad \Leftrightarrow \\ &\mathbf{b} + \boldsymbol{\Psi}_i \cdot \boldsymbol{\lambda}_i = 0 \quad \Leftrightarrow \\ &\mathbf{b} + \mathbf{L}_i^T \cdot \boldsymbol{\Sigma}_i \cdot \mathbf{L}_i \cdot \boldsymbol{\lambda}_i = 0 \end{aligned} \tag{4.3}$$

The solution for $\boldsymbol{\lambda}_i$ is now

$$\boldsymbol{\lambda}_i = -\mathbf{L}_i^T \cdot \boldsymbol{\Sigma}_i^{-1} \cdot \mathbf{L}_i \cdot \mathbf{b} \tag{4.4}$$

The inverse $\boldsymbol{\Sigma}_i^{-1}$ will be responsible for high values of $\boldsymbol{\lambda}_i$, if the eigenvalues tend to zero. We can set a lower bound and define the pseudo-inverse by

$$\boldsymbol{\Sigma}_i^{-1} = \begin{pmatrix} \sigma_{11}^{-1} & & & & \\ & \ddots & & & \\ & & \sigma_{jj}^{-1} & & \\ & & & \ddots & \\ & & & & \sigma_{MM}^{-1} \end{pmatrix} \tag{4.5}$$

Where the special extinction is given by

$$\sigma_{jj}^{-1} = \min\left(\frac{1}{\sigma_{jj}}, A\right) \tag{4.6}$$

A value frequently used for (normalized!) particle stencils is

$$A = 10^3 ... 10^4 \tag{4.7}$$

5. Improving Quality of the Differential Operators (Version 7)

Improving the quality of the differential operators is one of the biggest advantages of the FPM scheme in order to stabilize the simulations at critical locations. Stabilization can be achieved by two mayor algorithms.

1. Reduce the number of principal spatial directions if the point cloud is degenerated

2. Disturb slightly the right hand side $\mathbf{b}$ and gain a conveniently disturbed differential operator

For this chapter, let us make the following simplifications. Equation (2.10) suggests, that all the numerical differential operators to be produced are a linear combination of basis functions of the form

$$\mathbf{c}_i^* = -\mathbf{W}_i^2 \cdot \mathbf{K}_i \cdot \boldsymbol{\lambda}_i^* \tag{5.1}$$

In the method given in Equation (2.10), the basis functions are derived from the test functions $\mathbf{K}_i$ together with the weight functions $\mathbf{W}_i$. Here, we have the matrix of basis functions given by

$$\mathbf{B}_i = -\mathbf{W}_i^2 \cdot \mathbf{K}_i \tag{5.2}$$

Let us now suppose arbitrary basis functions $\mathbf{B}_i$. The columns of this matrix span the whole space out of which the differential operators $\mathbf{c}_i$ can be produced in the sense

$$\mathbf{c}_i = \mathbf{B}_i \cdot \boldsymbol{\alpha}_i \tag{5.3}$$

The $\boldsymbol{\alpha}_i$ are the coefficients of the linear combination of the columns of $\mathbf{B}_i$. Moreover, we have a space of test functions $\mathbf{T}_i$, for which we require certain (fixed) results of our linear operators, i.e.

$$\mathbf{T}_i \cdot \mathbf{c}_i^* = \mathbf{T}_i \cdot \mathbf{B}_i \cdot \boldsymbol{\alpha}_i^* = \mathbf{b}^* \tag{5.4}$$

Equation (5.4) is the abstract result of everything that was written up to now.

5.1. Reduction of principal number of spatial dimensions

The spatial reduction can be applied by a thickness analysis of the local stencil. Let us take the matrix $\mathbf{X}_i$, which is composed by the geometrical differences of the particle i to all of its neighbor particles j, i.e.

$$\mathbf{X}_i^{(.,j)} = \left(\mathbf{x}_j - \mathbf{x}_i\right)^T \tag{5.5}$$

$\mathbf{X}_i$ is called the geometrical basis of the particle i.

The list of distances of the neighbor particles with respect to a plane running through $\mathbf{x}_i$ and having the normal $\mathbf{n}$is given by

$$\mathrm{d} = \mathbf{X}_i \cdot \mathbf{n} \tag{5.6}$$

and the sum of all squared distances is

$$\mathbf{d}^T \cdot \mathbf{d} = \mathbf{n}^T \cdot \left(\mathbf{X}_i\right)^T \cdot \mathbf{X}_i \cdot \mathbf{n} \tag{5.7}$$

The minimization of $\mathbf{d}^T \cdot \mathbf{d}$ under the condition $\|\mathbf{n}\|_2 = 1$ leads to the eigenvalue and eigenvector problem

$$\left(\mathbf{X}_i\right)^T \cdot \mathbf{X}_i \cdot \mathbf{n} + \lambda \cdot \mathbf{n} = 0 \tag{5.8}$$

The eigenvectors $\mathbf{n}$ denote the direction, into which the local point cloud has maximal/minimal extensions. The corresponding eigenvalue represents the mean thickness of the pointcloud in this direction. An eigenvector, whose eigenvalue is too small compared to the biggest one, is called an invalid direction. Let the matrix $\mathbf{N}$ contain all valid eigenvectors (columnwise), then the matrix

$$\tilde{\mathbf{X}}_i = \mathbf{X}_i \cdot \mathbf{N} \tag{5.9}$$

is the valid geometrical basis of the particle i.
For the differential operators, it is wise to restrict the space of test functions to the valid geometrical basis. I.e. we have to multiply the matrix $\mathbf{T}_i$ with a permutation $\mathbf{P}_i$ such that the geometrical reduction is incorporated. The linear system to be solved instead of (5.4) is

$$\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{c}_i = \mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i \cdot \boldsymbol{\alpha}_i = \mathbf{P}_i \cdot \mathbf{b} = \hat{\mathbf{b}}_i \tag{5.10}$$

5.2. Quality gain by disturbing the right hand side

Let us now suppose we constructed the above linear system, out of which our differential operator follows simply by equation (5.3). Suppose furthermore, we have some quality criteria

$$Q\left(\mathbf{c}_i\right) < 0 \tag{5.11}$$

If the solution (5.10) does not fulfill the criteria of (5.11), we can now try to slightly modify the right hand side $\hat{\mathbf{b}}_i$ such that the norm of $\boldsymbol{\alpha}_i$ is minimized. So we try to solve the perturbed system

$$\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i \cdot \left(\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i\right) = \hat{\mathbf{b}}_i + \Delta\hat{\mathbf{b}}_i \tag{5.12}$$

Equivalently, we solve

$$\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i \cdot \Delta\boldsymbol{\alpha}_i = \Delta\hat{\mathbf{b}}_i \tag{5.13}$$

with the conditions

$$\Delta\hat{\mathbf{b}}_i^T \cdot \Delta\hat{\mathbf{b}}_i^T = \varepsilon^2 \tag{5.14}$$

and

$$\left(\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i\right)^T \cdot \left(\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i\right) \;=\; \min \tag{5.15}$$

Using Lagrange multipliers, this leads to

$$\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i + \sigma \cdot \left(\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i\right)^T \cdot \left(\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i\right) \cdot \Delta\boldsymbol{\alpha}_i \;=\; 0 \tag{5.16}$$

where σ is the (scalar) Lagrange multiplier.
We can first assume a small size of $\Delta\boldsymbol{\alpha}$ and find a solution (as a search direction!) by

$$\Delta\boldsymbol{\alpha}_i \;=\; -\left[\left(\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i\right)^T \cdot \left(\mathbf{P}_i \cdot \mathbf{T}_i \cdot \mathbf{B}_i\right)\right]^{-1} \cdot \boldsymbol{\alpha}_i \tag{5.17}$$

Finally we use the search direction $\Delta\boldsymbol{\alpha}_i$ to minimize the length of the vector $\boldsymbol{\alpha}_i$ and we correct alpha by

$$\boldsymbol{\alpha}_i := \boldsymbol{\alpha}_i + \tau \cdot \Delta\boldsymbol{\alpha}_i \tag{5.18}$$

where the τ is produced from the original $\boldsymbol{\alpha}_i$

$$\tau = \frac{\Delta\boldsymbol{\alpha}_i^T \cdot \boldsymbol{\alpha}_i}{\Delta\boldsymbol{\alpha}_i^T \cdot \Delta\boldsymbol{\alpha}_i} \tag{5.19}$$

The new alpha vector is now minimal under the given correction search direction $\Delta\boldsymbol{\alpha}_i$. One can now check if the new, resulting $\mathbf{c}_i$ comes closer to the quality conditions $Q\left(\mathbf{c}_i\right) < 0$. If not, the process starting from equation (5.12) can be repeated.

This whole procedure minimizes the operator vector $\mathbf{c}_i$ under minimally giving up parts of the right hand side vector $\hat{\mathbf{b}}_i$. We always have to remember, that by stabilization, we give up accuracy!

6. Conservative gradient formulation (Version 9)

In this chapter we will present an idea to construct discrete gradient operators with a conservation property.

We remember from chapter 1 that the approximation of some function $u(\mathbf{x})$ is given by

$$\tilde{u}(\mathbf{x}) = \sum c_j^0(\mathbf{x}) \cdot u_j = \left(\mathbf{c}^0(\mathbf{x})\right)^T \cdot \mathbf{u} \tag{5.20}$$

We remember, that
$\mathbf{c}^0(\mathbf{x})$ is the operator stencil established by least squares at the position $\mathbf{x}$,
$\mathbf{u}$ is the vector of all discrete function values at the particle locations,
$\tilde{u}(\mathbf{x})$ is as smooth as the weighting kernel $w(r)$.

Thus, if $\frac{d}{dr} w(r)$ is smooth, then $\tilde{u}(\mathbf{x})$ is smooth, and we can compute the derivative formally by

$$\frac{\partial}{\partial x} \tilde{u}(\mathbf{x}) = \left(\frac{\partial}{\partial x} \mathbf{c}^0(\mathbf{x}) \right)^T \cdot \mathbf{u} \tag{5.21}$$

The unknown derivatives $\frac{\partial}{\partial x} \mathbf{c}^0(\mathbf{x})$ can be produced by virtue of equations (2.9) and (2.10):

a) equation (2.9) lets us compute the derivative of the Lagrange multipliers

$$\begin{aligned} &\frac{\partial}{\partial x}\left(\mathbf{K}^T \cdot \mathbf{W}^T \cdot \mathbf{W} \cdot \mathbf{K}\right) \cdot \boldsymbol{\lambda}^0 + \mathbf{K}^T \cdot \mathbf{W}^T \cdot \mathbf{W} \cdot \mathbf{K} \cdot \frac{\partial}{\partial x}\left(\boldsymbol{\lambda}^0\right) = 0 \\ &\frac{\partial}{\partial x}\left(\boldsymbol{\lambda}^0\right) = -\left(\mathbf{K}^T \cdot \mathbf{W}^T \cdot \mathbf{W} \cdot \mathbf{K}\right)^{-1} \cdot \frac{\partial}{\partial x}\left(\mathbf{K}^T \cdot \mathbf{W}^T \cdot \mathbf{W} \cdot \mathbf{K}\right) \cdot \boldsymbol{\lambda}^0 \end{aligned} \tag{5.22}$$

b) equation (2.10) lets us compute the derivative of the stencil operator with respect to the Lagrange multipliers

$$\frac{\partial}{\partial x}\left(\mathbf{c}^0\right) = -\frac{\partial}{\partial x}\left(\mathbf{W}^2 \cdot \mathbf{K}\right) \cdot \boldsymbol{\lambda}^0 - \left(\mathbf{W}^2 \cdot \mathbf{K}\right) \cdot \frac{\partial}{\partial x}\left(\boldsymbol{\lambda}^0\right) \tag{5.23}$$

Plugging (5.22) into (5.23) gives

$$\frac{\partial}{\partial x}\left(\mathbf{c}^0\right)=-\frac{\partial}{\partial x}\left(\mathbf{W}^2\cdot\mathbf{K}\ \right)\cdot\boldsymbol{\lambda}^0+\left(\mathbf{W}^2\cdot\mathbf{K}\ \right)\cdot\left(\left(\mathbf{K}^T\cdot\mathbf{W}^T\cdot\mathbf{W}\ \cdot\mathbf{K}\ \right)^{-1}\cdot\frac{\partial}{\partial x}\left(\mathbf{K}^T\cdot\mathbf{W}^T\cdot\mathbf{W}\ \cdot\mathbf{K}\ \right)\cdot\boldsymbol{\lambda}^0\right) \tag{5.24}$$

The conservation properties are of continuous nature. Equation (5.21) suggests that Gauss' theorem holds, that is

$$\begin{aligned}\int_\Omega \nabla\tilde{u}\left(\mathbf{x}\right)\mathrm{d}\Omega\ &=\oint_{\partial\Omega}\tilde{u}\left(\mathbf{x}\right)\mathbf{n}\ \mathrm{d}\left(\partial\Omega\right)\\ \int_\Omega\sum_j\nabla c_j^0\left(\mathbf{x}\right)\cdot u_j\ \mathrm{d}\Omega\ &=\oint_{\partial\Omega}\sum c_j^0\left(\mathbf{x}\right)\cdot u_j\mathbf{n}\ \mathrm{d}\left(\partial\Omega\right)\end{aligned} \tag{5.25}$$

If we apply (5.25) to a delta function, established at the particle location i , then we find

$$\begin{aligned}\int_\Omega\sum_j\nabla c_j^0\left(\mathbf{x}\right)\cdot\delta_{ij}\ \mathrm{d}\Omega\ &=\oint_{\partial\Omega}\sum c_j^0\left(\mathbf{x}\right)\cdot\delta_{ij}\mathbf{n}\ \mathrm{d}\left(\partial\Omega\right)\\ \int_\Omega\nabla c_i^0\left(\mathbf{x}\right)\mathrm{d}\Omega\ &=\oint_{\partial\Omega}c_i^0\left(\mathbf{x}\right)\mathbf{n}\ \mathrm{d}\left(\partial\Omega\right)\end{aligned} \tag{5.26}$$

For any particle with the index i . Equation (5.26) helps us to find discrete entries for the particle volume V_i and boundary area A_i representations, that is

$$\begin{aligned}\int_\Omega\nabla c_i^0\left(\mathbf{x}\right)\mathrm{d}\Omega\ &=\oint_{\partial\Omega}c_i^0\left(\mathbf{x}\right)\mathbf{n}\ \mathrm{d}\left(\partial\Omega\right)\\ \sum_i\nabla c_j^0\left(\mathbf{x}_i\right)\cdot V_i\ &=\sum_i^{\partial\Omega}c_j^0\left(\mathbf{x}_i\right)\mathbf{n}_i A_i\\ \sum_i c_{ij}^{\nabla}\cdot V_i\ &=\sum_i^{\partial\Omega}c_{ij}^0\mathbf{n}_i A_i\end{aligned} \tag{5.27}$$

That means find V_i and A_i such that $\sum_i c_{ij}^{\nabla}\cdot V_i\ =\sum_i c_{ij}^0\mathbf{n}_i A_i$ holds for any index j . Use V_i and A_i for post-processing integrations of the domain or along the boundaries.

7. Test functions to be used in FPM

We have big experience in using monomials up to certain order as test functions.

The matrix of test functions (2.2) appears as

$$\mathbf{K}_i^T = \begin{pmatrix} 1 & \cdots & 1 & \cdots & 1 \\ (x_1 - x_i) & \cdots & (x_j - x_i) & \cdots & (x_N - x_i) \\ (y_1 - y_i) & \cdots & (y_j - y_i) & \cdots & (y_N - y_i) \\ (z_1 - z_i) & \cdots & (z_j - z_i) & \cdots & (z_N - z_i) \\ (x_1 - x_i)^2 & \cdots & (x_j - x_i)^2 & \cdots & (x_N - x_i)^2 \\ (x_1 - x_i)(y_1 - y_i) & \cdots & (x_j - x_i)(y_j - y_i) & \cdots & (x_N - x_i)(y_N - y_i) \\ (x_1 - x_i)(z_1 - z_i) & \cdots & (x_j - x_i)(z_j - z_i) & \cdots & (x_N - x_i)(z_N - z_i) \\ \vdots & & \vdots & & \vdots \\ & & & & \end{pmatrix} \tag{6.1}$$

Sometimes we would like to fulfill an additional condition on the central stencil entry c_{ii}^*. In this case, we have to add a line in the matrix above and obtain

$$\hat{\mathbf{K}}_i^T = \begin{pmatrix} \delta_{1i} & \cdots & \delta_{ji} & \cdots & \delta_{Ni} \\ 1 & \cdots & 1 & \cdots & 1 \\ (x_1 - x_i) & \cdots & (x_j - x_i) & \cdots & (x_N - x_i) \\ (y_1 - y_i) & \cdots & (y_j - y_i) & \cdots & (y_N - y_i) \\ (z_1 - z_i) & \cdots & (z_j - z_i) & \cdots & (z_N - z_i) \\ (x_1 - x_i)^2 & \cdots & (x_j - x_i)^2 & \cdots & (x_N - x_i)^2 \\ (x_1 - x_i)(y_1 - y_i) & \cdots & (x_j - x_i)(y_j - y_i) & \cdots & (x_N - x_i)(y_N - y_i) \\ (x_1 - x_i)(z_1 - z_i) & \cdots & (x_j - x_i)(z_j - z_i) & \cdots & (x_N - x_i)(z_N - z_i) \\ \vdots & & \vdots & & \vdots \\ & & & & \end{pmatrix} \tag{6.2}$$

With the modified matrix $\hat{\mathbf{K}}_i^T$, we solve the modified linear system

$$\hat{\mathbf{K}}_i^T \cdot \mathbf{c}_i = \hat{\mathbf{b}} \tag{6.3}$$

8. Right hand side vectors

Of special importance are the right hand side vectors. For certain operators, the right hand side vectors $\mathbf{b}$ as well as $\hat{\mathbf{b}}$ .are generally valid throughout the pointcloud.

8.1. Gradients

$$\begin{aligned}
\mathbf{b}^x &= (\,0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad) \\
\mathbf{b}^y &= (\,0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad) \\
\mathbf{b}^z &= (\,0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad \cdots \quad)
\end{aligned} \tag{7.1}$$

Additionally, we could require that the central element of the stencil vanish. So we have

$$\begin{aligned}
\hat{\mathbf{b}}^x &= (\,0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad) \\
\hat{\mathbf{b}}^y &= (\,0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad \cdots \quad) \\
\hat{\mathbf{b}}^z &= (\,0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad \cdots \quad)
\end{aligned} \tag{7.2}$$

This requirement makes sense, it amounts to requiring

$$\tilde{\nabla}\left(\delta\left(\mathbf{x}_i\right)\right)_i \overset{!}{=} 0 \tag{7.3}$$

8.2. Laplaciens

$$\mathbf{b}^{\Delta} = (\,0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 2 \quad 0 \quad 2 \quad 0 \quad \cdots) \tag{7.4}$$

As well as

$$\hat{\mathbf{b}}^{\Delta} = \left(-\frac{\alpha}{h_i^2} \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 2 \quad 0 \quad 2 \quad 0 \quad \cdots\right) \tag{7.5}$$

Here we try to set the certain Laplacian value for the delta function at the central stencil entry by

$$\tilde{\Delta}\left(\delta\left(\mathbf{x}_i\right)\right)_i \overset{!}{=} -\frac{\alpha}{h_i^2} \tag{7.6}$$

The order of magnitude of α is 5…15.

8.2.1. Achieving most possible diagonal dominance

For all implicit computations in FPM (for example pressure Poisson equations, heat transport/conduction equations), we form sparse linear systems out of the local differential operators. In order to solve the sparse linear systems, the operators should show the possibly best diagonal dominance. That means for a local stencil, that c_{ii}^{Δ} is possibly big compared to the non-diagonal entries c_{ij}^{Δ}. One suggestion is to minimize the functional

$$g \equiv \frac{\sum_{j=1}^{N}\left(c_{ij}^{\Delta}\right)^2}{\left(c_{ii}^{\Delta}\right)^2} = \frac{\left(\mathbf{c}_i^{\Delta}\right)^T \cdot \mathbf{c}_i^{\Delta}}{\left(c_{ii}^{\Delta}\right)^2} \tag{7.7}$$

Suppose we have computed a local Laplacien operator $\hat{\mathbf{c}}_i^{\Delta}$ by setting

$$\hat{\mathbf{b}}^{\Delta} = \begin{pmatrix} A & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 2 & 0 & \cdots \end{pmatrix} \tag{7.8}$$

with some arbitrary value $c_{ii}^{\Delta} = A$. The whole stencil will change if A is changed, i.e. we must have an expression $\frac{\partial \mathbf{c}_i^{\Delta}}{\partial A}$. How can we compute it? We remember from equation (2.10) that

$$\mathbf{c}_i^{\Delta} = \mathbf{W}_i^2 \cdot \hat{\mathbf{K}}_i \cdot \left(\hat{\mathbf{K}}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \hat{\mathbf{K}}_i\right)^{-1} \cdot \hat{\mathbf{b}}^{\Delta} \tag{7.9}$$

We apply the derivative with respect to A and find

$$\mathbf{d}_i^{\Delta} \equiv \frac{\partial \mathbf{c}_i^{\Delta}}{\partial A} = \mathbf{W}_i^2 \cdot \hat{\mathbf{K}}_i \cdot \left(\hat{\mathbf{K}}_i^T \cdot \mathbf{W}_i^T \cdot \mathbf{W}_i \cdot \hat{\mathbf{K}}_i\right)^{-1} \cdot \frac{\partial \hat{\mathbf{b}}^{\Delta}}{\partial A} \tag{7.10}$$

together with the facts

$$\frac{\partial \hat{\mathbf{b}}^{\Delta}}{\partial A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \end{pmatrix} \tag{7.11}$$

and

$$d_{ii}^{\Delta} = \frac{\partial c_{ii}^{\Delta}}{\partial A} = 1 \tag{7.12}$$

The functional g in (7.7) can now be minimized analytically.

$$\frac{\partial g}{\partial A} = 2\left\langle \left(\mathbf{c}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \cdot \left(c_{ii}^{\Delta}\right)^{-2} - 2\left\langle \left(\mathbf{c}_i^{\Delta}\right)^T \cdot \mathbf{c}_i^{\Delta} \right\rangle \cdot \left(c_{ii}^{\Delta}\right)^{-3} \cdot d_{ii}^{\Delta} \overset{!}{=} 0 \tag{7.13}$$

In other words,

$$\left\langle \left(\mathbf{c}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \cdot \left(c_{ii}^{\Delta}\right) - \left\langle \left(\mathbf{c}_i^{\Delta}\right)^T \cdot \mathbf{c}_i^{\Delta} \right\rangle \cdot d_{ii}^{\Delta} \overset{!}{=} 0 \tag{7.14}$$

The solution $\mathbf{c}_i^{\Delta}$ will be assembled out of an already computed stencil and a correction of the form

$$\mathbf{c}_i^{\Delta} = \widehat{\mathbf{c}}_i^{\Delta} + \alpha \mathbf{d}_i^{\Delta} \tag{7.15}$$

which is to be plugged into (7.14). Therewith, we obtain

$$\left\langle \left(\widehat{\mathbf{c}}_i^{\Delta} + \alpha \mathbf{d}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \cdot \left(\widehat{c}_{ii}^{\Delta} + \alpha\right) - \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta} + \alpha \mathbf{d}_i^{\Delta}\right)^T \cdot \left(\widehat{\mathbf{c}}_i^{\Delta} + \alpha \mathbf{d}_i^{\Delta}\right) \right\rangle \cdot \overset{!}{=} 0 \tag{7.16}$$

Equivalent changes lead to

$$\left(\left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle + \alpha \left\langle \left(\mathbf{d}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle\right) \cdot \left(\widehat{c}_{ii}^{\Delta} + \alpha\right) = \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \widehat{\mathbf{c}}_i^{\Delta} \right\rangle + 2\alpha \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle + \alpha^2 \left\langle \left(\mathbf{d}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle$$

$$\left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \widehat{c}_{ii}^{\Delta} + \alpha \left\langle \left(\mathbf{d}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \widehat{c}_{ii}^{\Delta} + \alpha \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle = \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \widehat{\mathbf{c}}_i^{\Delta} \right\rangle + 2\alpha \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle$$

$$\left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \widehat{c}_{ii}^{\Delta} - \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \widehat{\mathbf{c}}_i^{\Delta} \right\rangle = \alpha \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle - \alpha \left\langle \left(\mathbf{d}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \widehat{c}_{ii}^{\Delta}$$

and finally to

$$\frac{\left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \widehat{c}_{ii}^{\Delta} - \left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \widehat{\mathbf{c}}_i^{\Delta} \right\rangle}{\left\langle \left(\widehat{\mathbf{c}}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle - \left\langle \left(\mathbf{d}_i^{\Delta}\right)^T \cdot \mathbf{d}_i^{\Delta} \right\rangle \widehat{c}_{ii}^{\Delta}} = \alpha \tag{7.17}$$

Equation (7.17) shows, that we need to compute an arbitrary stencil $\widehat{\mathbf{c}}_i^{\Delta}$ together with $\mathbf{d}_i^{\Delta}$, both item can be computed simultaneously. As they are

computed with the same matrix, only with different right hand side, the computational effort does not rise significantly. Out of the two items, the optimized stencil is computed by equation (7.15).

Equation (7.17) is not only valid for Laplacians. Similar considerations can be made for diagonal-optimizing any operator.

8.3. Operator for Neumann boundary conditions

For diffusion equations, it is necessary to set forth Neumann-type boundary conditions, i.e. we would like to set the directional derivative of some function

$$\frac{\partial u}{\partial n} \tag{7.18}$$

Here, one can use the right hand side vector

$$\frac{\partial \hat{\mathbf{b}}^{\partial_n}}{\partial A} = \left(\frac{\beta}{h_i} \quad n^x \quad n^y \quad n^z \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \right) \tag{7.19}$$

The value β is in the order of magnitude 5…10.

Diagonal dominance considerations are made here similarly to the Laplace operator.

9. Diffusive Transport Operator with variable diffusion coefficient

Let us have a look at the operator

$$\nabla^T \left(\eta (\nabla u) \right) \tag{8.1}$$

where we assume, that η might strongly change in space.

Under given coordinate transformation we have

$$\nabla u \equiv \mathbf{M}^T \cdot \bar{\nabla} u \tag{8.2}$$

and primitive differentiation given by

$$\frac{\partial u}{\partial x_m} \equiv \mathbf{M}_{(*,m)} \cdot \bar{\nabla} u \tag{8.3}$$

$$\nabla^T \left(\eta(\nabla u)\right) = \nabla^T \left(\eta \cdot \mathbf{M}^T \cdot \bar{\nabla} u\right) = \bar{\nabla}^T \cdot \mathbf{M} \cdot \left(\eta \cdot \mathbf{M}^T \cdot \bar{\nabla} u\right) = \bar{\nabla}^T \cdot \left(\mathbf{M} \cdot \eta \cdot \mathbf{M}^T \cdot \bar{\nabla} u\right) \tag{8.4}$$

So we have

$$\begin{aligned}
&\nabla^T \left(\eta(\nabla u)\right) = \bar{\nabla}^T \cdot \left(\mathbf{M} \cdot \eta \cdot \mathbf{M}^T \cdot \bar{\nabla} u\right) = \\
&\left(\eta \cdot \boldsymbol{\mathcal{M}}_{11} \cdot u_{\bar{x}}\right)_{\bar{x}} + \left(\eta \cdot \boldsymbol{\mathcal{M}}_{12} \cdot u_{\bar{y}}\right)_{\bar{x}} + \left(\eta \cdot \boldsymbol{\mathcal{M}}_{13} \cdot u_{\bar{z}}\right)_{\bar{x}} + \\
&\left(\eta \cdot \boldsymbol{\mathcal{M}}_{21} \cdot u_{\bar{x}}\right)_{\bar{y}} + \left(\eta \cdot \boldsymbol{\mathcal{M}}_{22} \cdot u_{\bar{y}}\right)_{\bar{y}} + \left(\eta \cdot \boldsymbol{\mathcal{M}}_{23} \cdot u_{\bar{z}}\right)_{\bar{y}} + \\
&\left(\eta \cdot \boldsymbol{\mathcal{M}}_{31} \cdot u_{\bar{x}}\right)_{\bar{z}} + \left(\eta \cdot \boldsymbol{\mathcal{M}}_{32} \cdot u_{\bar{y}}\right)_{\bar{z}} + \left(\eta \cdot \boldsymbol{\mathcal{M}}_{33} \cdot u_{\bar{z}}\right)_{\bar{z}}
\end{aligned} \tag{8.5}$$

where

$$\boldsymbol{\mathcal{M}} = \mathbf{M} \cdot \mathbf{M}^T$$

Consequently,

$$\begin{aligned}
&\nabla^T \left(\eta(\nabla u)\right) = \\
&\eta_{\bar{x}} \cdot \left(\boldsymbol{\mathcal{M}}_{11} \cdot u_{\bar{x}} + \boldsymbol{\mathcal{M}}_{12} \cdot u_{\bar{y}} + \boldsymbol{\mathcal{M}}_{13} \cdot u_{\bar{z}}\right) + \\
&\eta_{\bar{y}} \cdot \left(\boldsymbol{\mathcal{M}}_{21} \cdot u_{\bar{x}} + \boldsymbol{\mathcal{M}}_{22} \cdot u_{\bar{y}} + \boldsymbol{\mathcal{M}}_{23} \cdot u_{\bar{z}}\right) + \\
&\eta_{\bar{z}} \cdot \left(\boldsymbol{\mathcal{M}}_{31} \cdot u_{\bar{x}} + \boldsymbol{\mathcal{M}}_{32} \cdot u_{\bar{y}} + \boldsymbol{\mathcal{M}}_{33} \cdot u_{\bar{z}}\right) + \\
&\eta \cdot \left(\boldsymbol{\mathcal{M}}_{11} \cdot u_{\bar{x}\bar{x}} + \boldsymbol{\mathcal{M}}_{12} \cdot u_{\bar{y}\bar{x}} + \boldsymbol{\mathcal{M}}_{13} \cdot u_{\bar{z}\bar{x}} + \boldsymbol{\mathcal{M}}_{21} \cdot u_{\bar{x}\bar{y}} + \boldsymbol{\mathcal{M}}_{22} \cdot u_{\bar{y}\bar{y}} + \boldsymbol{\mathcal{M}}_{23} \cdot u_{\bar{z}\bar{y}} + \boldsymbol{\mathcal{M}}_{31} \cdot u_{\bar{x}\bar{z}} + \boldsymbol{\mathcal{M}}_{32} \cdot u_{\bar{y}\bar{z}} + \boldsymbol{\mathcal{M}}_{33} \cdot u_{\bar{z}\bar{z}}\right)
\end{aligned} \tag{8.6}$$

10. Conservation properties of Laplace operator

For some function u the Laplacien together with the Neumann operator should hold

$$\int_{\Omega} \Delta u \cdot dV + \oint_{\partial\Omega} \frac{\partial u}{\partial n} \cdot dA = 0 \tag{8.7}$$

Obeying the normal convention in MESHFREE (inward pointing normals). These integral can be approximated in the sense

$$\sum_{i \in \Omega} \Delta u_i \cdot V_i + \sum_{i \in \partial\Omega} \left. \frac{\partial u}{\partial n} \right|_i \cdot A_i \approx 0 \tag{8.8}$$

Let us replace the Laplacien and Neuman with with their appropriate approximation

$$\sum_{i \in \Omega} \sum_j c_{ij}^{\Delta} \left(u_j - u_i \right) \cdot V_i + \sum_{i \in \partial\Omega} \sum_j c_{ij}^{\partial/\partial n} \left(u_j - u_i \right) \cdot A_i \approx 0 \tag{8.9}$$

Let us consider a delta function at a fixed location j, then this equation simplifies to

$$c_j^{cons} \equiv \sum_{i \in \Omega} c_{ij}^{\Delta} \cdot V_i + \sum_{i \in \partial\Omega} c_{ij}^{\partial/\partial n} \cdot A_i \approx 0 \quad \text{for fixed } j \tag{8.10}$$

c_j^{cons} is the conservation property of the Laplace/Neumann operators in MESHFREE. It is hopefully very small.