Metric construction based on error estimation

This document presents the metric matrix construction method based on error estimation for anisotropic adaptive meshing. More details are available in the article of Thierry Coupez. This method is implemented in *Cimlib* in 2 models: *ISMaTC* and *SsfMaTC* (TC denotes the name of author), which trait the different procedures of the calculation of metric matrix.

Some useful notations are given below:

Notations	Definitions
d	space dimension
N	number of scalar field that the mesh adaptation based
N_e	given number of edges
N_E	given number of elements
$\mathcal{K}$	set of elements
$\mathcal{N}$	set of nodes
$K \in \mathcal{K}$	mesh element
$\mathbf{X}^i, i \in \mathcal{N}$	vector of coordinates for the i^{th} node
$\mathbf{X}^{ij} = \mathbf{X}^j - \mathbf{X}^i$	edge vector made of nodes i and j sharing at least one element
$h_{ij} = \lvert\mathbf{X}^{ij}\rvert$	edge length
$\Gamma(i) = \{j \in \mathcal{N}, \exists K \in \mathcal{K}, \mathbf{X}^{ij} \in K\}$	set of nodes connected to node i
$\lvert\Gamma(i)\rvert$	cardinality of $\Gamma(i)$

1 Principle

The continuous metric field defined at the mesh node i is obtained from:

$$\mathbb{M}^i = \left(\frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1} \tag{1}$$

where s_{ij} is the stretching factor defined by the global error λ and the local edge error e_{ij}

$$s_{ij} = \min \left(\left(\frac{\lambda}{e_{ij}} \right)^{\frac{1}{p+2}}, \frac{h_{ij}}{h_{\min}} \right) \tag{2}$$

where p is a coefficient which is taken 1.5, $h_{\min}$ the minimal mesh size set by user and λ is calculated by:

$$\lambda = \left(\frac{\sum_i \sum_{j \in |\Gamma(i)|} e_{ij}^{\frac{p}{p+2}}}{N_e} \right)^{\frac{p+2}{p}} \tag{3}$$

where N_e is the number of edges calculated by $N_e = N_E D(D-1)$. Here D is the number of nodes of an element and N_E is a given number of elements in input parameters. Note that the sum is firstly done with all edges connected to node i and then with all nodes.

The local edge error is taken by:

$$e_{ij} = \max\left(|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|, \epsilon_{\min}|\mathbf{X}^{ij}|^2\right) \tag{4}$$

where $\epsilon_{\min}$ is a coefficient set by user to control the global mesh size and $\mathbf{G}^{ij} = \mathbf{G}^j - \mathbf{G}^i$ with $\mathbf{G}^i$ the gradient defined at the node i which will be detailed below.

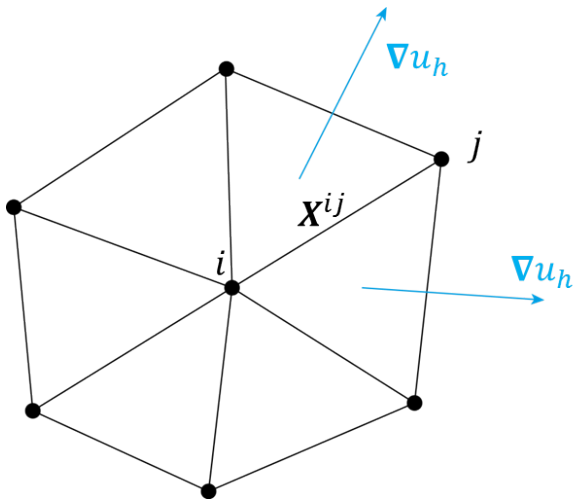


Figure 1: ∇u_h (case scalar) of each element

Consider that we have a $P1$ field $\mathbf{u}$ (composed of N scalar field u_k, $k \in [1, N]$), then the gradient of this filed $\nabla \mathbf{u}_h$ (of element K) is $P0$ (Fig.1), which is discontinuous from element to element but its projection on its edge is continuous because:

$$\nabla \mathbf{u}_h \cdot \mathbf{X}^{ij} = \mathbf{U}^j - \mathbf{U}^i = \mathbf{U}^{ij} \quad \left(\mathbf{X}^{ij} \in K\right) \tag{5}$$

where $\mathbf{U}^i$ is the nodal value at node i.

The objective is to find a gradient field at node which minimize the total error of edges connected to this node i. It means:

$$\mathbf{G}^i = \arg\min_{\mathbf{G}} \left(\sum_{j \in \Gamma(i)} |(\mathbf{G} - \nabla \mathbf{u}_h) \cdot \mathbf{X}^{ij}|^2 \right) = \arg\min_{\mathbf{G}} \left(\sum_{j \in \Gamma(i)} |(\mathbf{G} \cdot \mathbf{X}^{ij} - \mathbf{U}^{ij}|^2 \right) \tag{6}$$

Taking the derivative of Eq.6 equal to 0 with respect to $\mathbf{G}$, we can get:

$$\mathbf{G}^i = \left(\underbrace{\sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}}_{\mathbf{A}} \right)^{-1} \left(\underbrace{\sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \mathbf{U}^{ij}}_{\mathbf{B}} \right) \tag{7}$$

2 Calculating procedures of metric matrix

The calculation of metric matrix is divided into several procedures done with the model *SsfMaTC* and *ISMaTC* in *Cimlib*. All operations over edges are handled by *SsfMaTC* while all operations over nodes are handled by *ISMaTC*. The flow chart below shows how it works.

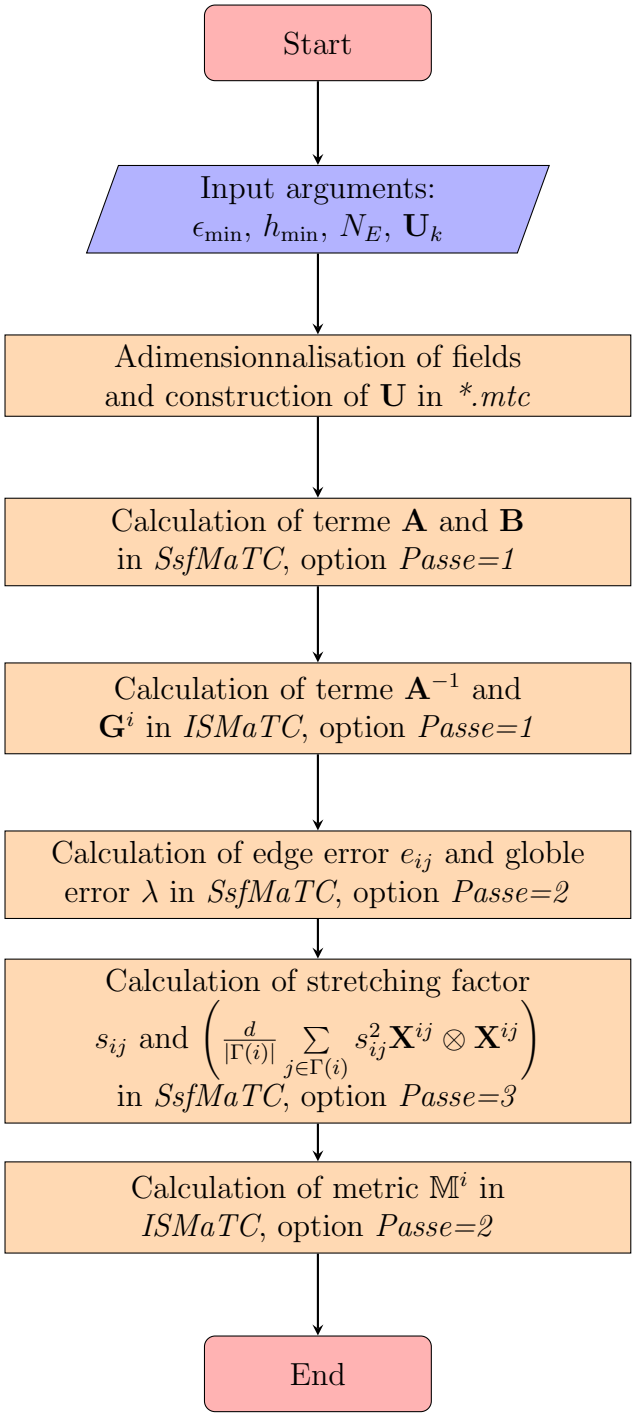


Figure 2: Calculating procedures of metric matrix

3 Construction of U and calculation of e_{ij}

Suppose that we have 3 ($N = 3$) scalar variable fields: Heaviside function $\mathcal{H}$, liquid fraction g^l and temperature T in 3D case ($d = 3$). Then the adimensionnalisation can be done by a function f, for example:

$$\mathcal{H}' = f(\mathcal{H}), \quad (\mathcal{H}_{min}, \mathcal{H}_{max}) \rightarrow (0, 1) \tag{8}$$

where $\mathcal{H}_{min}$ and $\mathcal{H}_{max}$ are two truncation limits. This is illustrated in Fig.3.

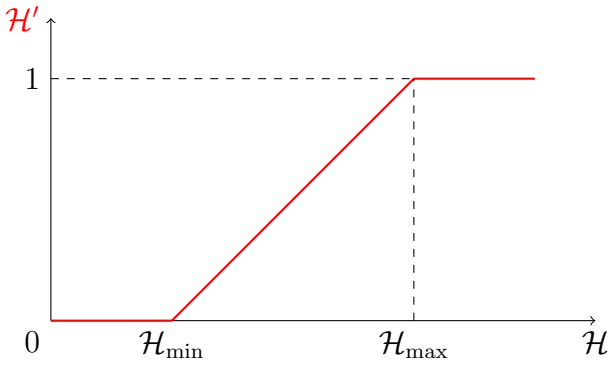


Figure 3: Adimensionnalisation of a field

$\mathcal{H}_{min}$ and $\mathcal{H}_{max}$ can be taken as $\min(\mathcal{H})$ and $\max(\mathcal{H})$ respectively. However, sometimes we need to remesh in a certain variable interval. Another choice can be taken:

$$\mathcal{H}_{\text{min}} = \alpha \mathcal{H}_{\text{mean}} \quad \mathcal{H}_{\text{max}} = \frac{1}{\alpha} \mathcal{H}_{\text{mean}} \quad \text{with}\, \alpha < 1 \tag{9}$$

where $\mathcal{H}_{\text{mean}}$ is the mean value of $\mathcal{H}$ and α is set by user.

Once the adimensional fields are obtained and then weighted ($\mathcal{H}$, g^l and T are used to represent these fields in order to simplify the notation), we can construct the vector $\mathbf{U}$ (size $1 \times N$) by:

$$\mathbf{U} = \begin{bmatrix} \mathcal{H} & g^l & T \end{bmatrix} \tag{10}$$

Then $\mathbf{G}^i$ can be obtained with Eq.7 and it is a matrix of size $d \times N$ composed of:

$$\mathbf{G}^i = \begin{bmatrix} \mathbf{G}^i_1 & \mathbf{G}^i_2 & \mathbf{G}^i_3 \end{bmatrix} \tag{11}$$

where $\mathbf{G}^i_k$ (size $d \times 1$, $k \in [1, N]$) denotes the gradient of $\mathcal{H}$, g^l and T at node i. Then $\mathbf{G}^{ij} = \mathbf{G}^j - \mathbf{G}^i$ has the same size as $\mathbf{G}^i$.

In order to obtain the edge error in Eq.4, we should at first calculate:

$$\begin{aligned} \mathbf{G}^{ij} \cdot \mathbf{X}^{ij} &= \begin{bmatrix} \mathbf{G}^{ij}_1 & \mathbf{G}^{ij}_2 & \mathbf{G}^{ij}_3 \end{bmatrix} \cdot \mathbf{X}^{ij} \\ &= \begin{bmatrix} \mathbf{G}^{ij}_1 \cdot \mathbf{X}^{ij} & \mathbf{G}^{ij}_2 \cdot \mathbf{X}^{ij} & \mathbf{G}^{ij}_3 \cdot \mathbf{X}^{ij} \end{bmatrix} \end{aligned} \tag{12}$$

which has the size of $1 \times N$. It includes the error contribution of each field. Then $|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|$ can be calculated by several methods, which is coded in *SsfMaTC, Passe=1* is:

$$|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| = \sqrt{\sum_{k=1}^{N} \left(\mathbf{G}^{ij}_k \cdot \mathbf{X}^{ij} \right)^2} \tag{13}$$

It means that the error contribution is accumulated. Another choice is to take the maximum of all the error contributions:

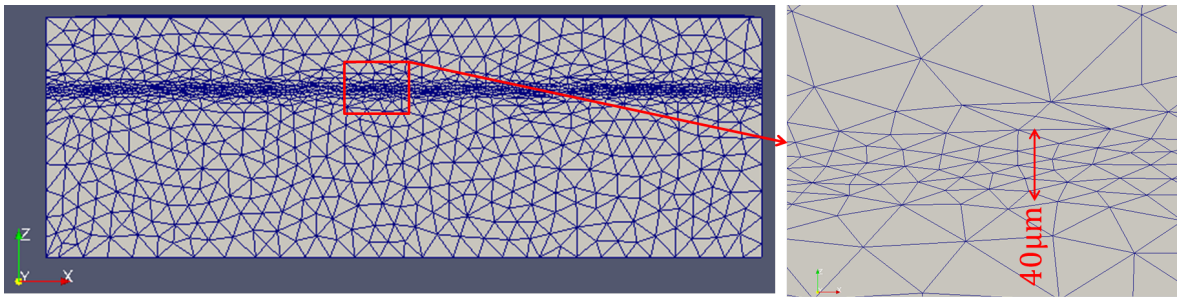
$$|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| = \max_k |\mathbf{G}_k^{ij} \cdot \mathbf{X}^{ij}| \tag{14}$$

These two choice will be compared.

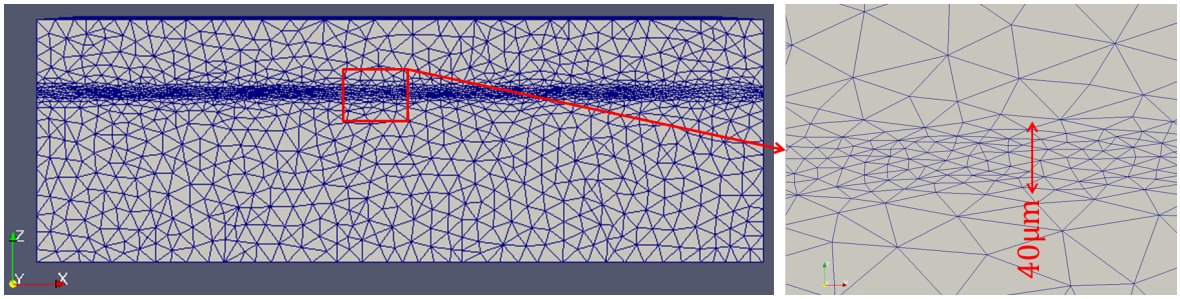
4 Examples

3 tests are done to investigate the input parameters $\epsilon_{\min}$ and N_E. Results in Fig.4a and 4b show that with more given elements, mesh size gets smaller in all the domain. However, neither of them attends the desired $h_{\min}$ around the interface.

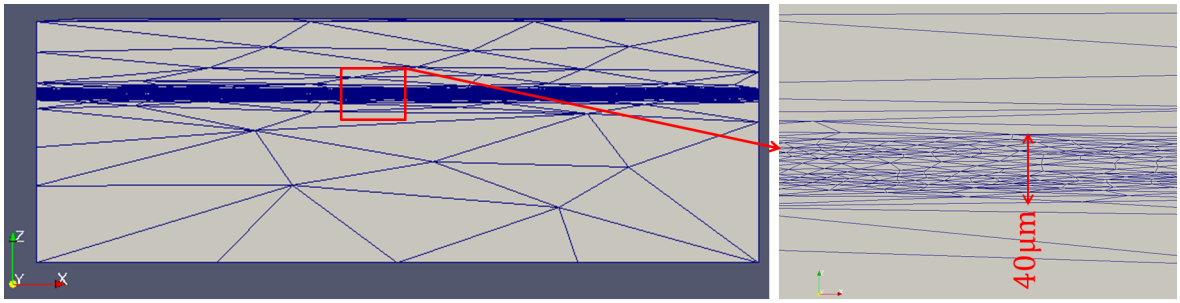
The coefficient $\epsilon_{\min}$ is used to control the global mesh size. From Eq.4 we know that $e_{ij} \uparrow$ when $\epsilon_{\min} \uparrow$, so $s_{ij} \downarrow$, consequently the global mesh size becomes smaller. We can find this by comparing Fig.4b and 4c. On the other hand, the mesh is refined in normal direction around the interface ($h_{\min}$ is attended) but stretched in the tangent direction, because the mesh size is determined by the Heaviside function and it is dominant in normal direction.



(a) $\epsilon_{\min} = 3 \times 10^6$, $N_E = 200000$, real element number = 172238



(b) $\epsilon_{\min} = 3 \times 10^6$, $N_E = 500000$, real element number = 423127



(c) $\epsilon_{\min} = 3 \times 10^3$, $N_E = 500000$, real element number = 387638

Figure 4: Domain $3 \times 1 \times 1\,\mathrm{mm}^3$, $h_{\min} = 4\,\mu\mathrm{m}$, remeshing based on Heaviside function