1-Dimensional Meshing Algorithm for ABI

Liam Fisher

June 12, 2024

1 Introduction

The goal of this exercise is to mesh a 1D domain of fixed length L bounded at either end by elements of known length s_1 and s_2. The aim is to specify the number and length of elements within this interval under the condition that "for best results, computational models gradually change element sizes across the model". Notably, an explicit definition of "gradually" is not provided.

2 Method

2.1 Defining Gradual Change

The measurement of change between adjacent elements is taken to be the ratio of their respective lengths. If we divide the interval L into n elements $L_1, L_2, ..., L_n$ (the value of L_i is the length of the i^{th} element) then we can define a set of ratios $R_1, R_2, ..., R_n$ such that $L_1R_1 = L_2, L_2R_2 = L_3, etc.$ Notably, at the beginning and endpoint of the interval $s_1R_0 = R_1$ and $L_nR_n = s_2$ respectively.

The value of an individual R_i representing maximally gradual change is 1, i.e. when two adjacent elements have the same length. The goal is to have maximally gradual change over the entire interval L, which requires minimising some norm over the set $R_0...R_n$. Notably, the goal is to minimise a form of $|R - 1|$, not merely minimising R, as a small R-value of e.g. 0.1 would represent a large difference in size of adjacent elements.

- Minimising a summation of $|R - 1|_{0...n}$ would not be appropriate because it penalizes increasing the number of elements.
- Minimising an average value of $|R - 1|_{0...n}$ would not be appropriate, because this could be achieved by meshing L with many tiny elements such that large $|1 - R_0|$ and $|1 - R_n|$ values exist at the boundaries of the interval, outweighed by the many small identically sized elements in the middle of the interval with $R = 1$.
- The most sensible objective is to minimise $|R - 1|_{max}$

Note: It was also explicitly specified that "there shouldn't be any elements larger than the largest size at either end", which is taken to mean that $L_i \leq max(s_1, s_2)$ for all $i \in n$.

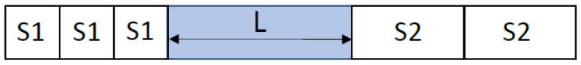

Figure 1: An illustration of the problem.

2.2 Mesh Generation

Consider the case where $n = 2$, a mesh with 2 elements and 3 values of R. Decreasing the width of L_1 necessarily increases the width of L_2, as the mesh must always fill the interval L. We have:

$$\begin{aligned} s_1 R_0 &= L_1 \\ L_1 R_1 &= L_2 \\ L_2 R_2 &= s_2 \\ L_1 + L_2 &= L \end{aligned}$$

We can write each R in terms of L_1:

$$\begin{aligned} R_0 &= L_1/s_1 \\ R_1 &= (L - L_1)/L_1 \\ R_2 &= s_2/(L - L_1) \end{aligned}$$

In this two-element case the objective function $|R-1|_{max}$ is only dependent on L_1, and this can therefore be treated as a univariate minimisation problem.

However, upon adding an additional element:

$$\begin{aligned} s_1 R_0 &= L_1 \\ L_1 R_1 &= L_2 \\ L_2 R_2 &= L_3 \\ L_3 R_3 &= s_2 \\ L_1 + L_2 + L_3 &= L \end{aligned}$$

We can similarly attempt to simplify this:

$$\begin{aligned} s_1 R_0 &= L_1 \\ s_1 R_0 R_1 &= L_2 \\ s_1 R_0 R_1 R_2 &= L_3 \\ s_1 R_0 R_1 R_2 R_3 &= s_2 \\ L_1 + L_2 + L_3 &= L \end{aligned}$$

With the extra element, there is no longer a way to express $|R-1|_{max}$ as a function of 1 variable, and this becomes a much more difficult situation to optimize.

However, we can enforce the condition that all $R > 1$ or all $R < 1$, as any mesh with successive element sizes decreasing then increasing (or vice versa) from s_1 to s_2 can be "smoothed" to either a pure increase or a pure decrease. (If $s_1 < s_2$ then all $R_i > 1$, else $s_1 > s_2$ and all $R_i < 1$.

Additionally, we want each R-value to be as close to 1 as possible for maximum smoothness, and this choice of cost function $|R-1|_{max}$ makes it difficult to take advantage of significant variation in R-values.

Therefore, rather than attempting to optimize each $R_0, R_1...$ separately, we make a simplifying assumption that $R_0 = R_1... = R_F$, where R_F is some value 'close to' 1.

The final equation above (for the 3-element case, $L_1 + L_2 + L_3 = L$) can therefore be written as:

$$s_1 R_F + s_1 R_F^2 + s_1 R_F^3 = L$$

This is a univariate polynomial with indeterminate R_F and is straightforward to solve numerically even for arbitrarily large n. (E.g. by Newton's Method with an initial guess of $R_F = 1$).

However, this formulation does not account for s_2 or R_n. An unoptimized R_n at the end of the L-interval could significantly penalise the cost function, i.e. it represents a potential loss of mesh quality. Using a sensible choice of n provides some amount of indirect control over the value of L_n and therefore the value of R_n. Furthermore, this "forward mesh" technique can be complemented by an analogous "backward mesh", which conversely defines the element lengths as scaled values of s_2 and ignores s_1 and R_0:

$$s_2 R_B^3 + s_2 R_B^2 + s_2 R_B = L$$

This can separately be solved numerically to find the value of R_B.

Each of these techniques ("forward" and "backward") generates a mesh of elements $L_1, L_2, ..., L_n$. A final mesh can be generated by taking the average length of each respective element.

To summarise a general formulation for n elements:

$$\sum_{i=1}^{n} s_1 R_F^i = L$$

This polynomial can be solved numerically for R_F, to produce a 'forward' mesh of elements $L_{1...n}$:

$$L_{i forward} = s_1 R_F^i, i \in 1...n$$

A 'backward' mesh can be generated by a similar method:

$$\sum_{i=1}^{n} s_2 R_B^i = L$$

$$L_{i backward} = s_2 R_B^{n+1-i}, i \in 1...n$$

A final mesh can be generated by averaging the lengths of each respective element from the forward and backward meshes:

$$L_{i(final)} = \frac{L_{i(forward)} + L_{i(backward)}}{2}, i \in 1...n$$

2.3 Measuring Smoothness $|R-1|_{max}$ for the Final Mesh

Using the example of a 2-element mesh where the second element is 1.5× the width of the first element:

$$L_2 = 1.5 \times L_1$$

If R is defined as $R = \frac{L_2}{L_1}$ then $|R-1|$ for this element boundary is $|1.5-1| = 0.5$. However, if a new mesh is created with the same two elements in reverse order:

$$L_2 \times 1.5 = L_1$$

Then by the prior definition of $R = \frac{L_2}{L_1}$, the $|R-1|$ measure for this element boundary is $|\frac{1}{1.5} - 1| = \frac{1}{3}$. The overall 'mesh smoothness' measure in each of these cases should not be different, because each mesh is only a mirror of the other. Therefore, when taking $|R-1|_{max}$ as a measure of the overall mesh quality, R-values must be generated by both comparing elements left-to-right and right-to-left, or by measuring R_i on elements pre-processed such that $L_i < L_{i+1}$.

2.4 Choosing the value of n

The method described above generates a mesh on the interval L for a specific number of elements n.

The initial guess for n is:

$$n = \left\lfloor \frac{L}{\frac{1}{2}(s_1 + s_2)} \right\rceil$$

Where $\lfloor \bullet \rceil$ is a function that rounds its argument to the nearest integer.
Essentially, this 'guess' is based on the number of elements that would approximately fill the interval L with identical elements that are the average width of s_1 and s_2. For the trivial case that $s_1 = s_2$ and $\frac{L}{s} \in Z$ this n corresponds to an exact solution with $|R-1|_{max} = 0$. For the general case where $s_1 \neq s_2$,

elements at one end of the mesh will decrease in size while elements at the other end increase, thus preserving low $|R-1|_{max}$ and maintaining the total width of the mesh without altering this initial n estimate.

For additional accuracy, n is adjusted by performing an iterative search in the neighbourhood of the original n estimate, (assuming convexity of $|R-1|_{max}$ as a function of n), stopping when $|R-1|_{max}$ no longer decreases.