1 METHOD OF MANUFACTURED SOLUTIONS

For completeness, we define two specific problems:

- MFP-1: Reach a Steady-State Solution from Zero.
- MFP-2: Unsteady-State for All Times.

For the above we give explicit forcing terms, boundary and initial conditions. They are derived departing from a known solution

$$u_e(x,t) = \tilde{f}(t)\tilde{u}(x), \qquad (1)$$

which we trim according to the implementations we want to exercise. Although this step could be seen like a loss of time, since we are not dealing with an actual problem of industrial or clinical application, it has many benefits. It gives us a known benchmark against which we can compare numerical solutions. This can not only help us find bugs in the implementation, but also certify the expected convergence rates in terms of mesh and timestep sizes. Later on, when we deal with an equation whose solution is unknown, we can be assured that the code we have written actually solves the equations written out on paper.

To simplify our equations, we chose $\tilde{u}(x)$ such that

$$\frac{\partial^2 \tilde{u}}{\partial x^2} = C. \qquad (2)$$

For $\delta \in \mathbb{R}$, this can be achieved with

$$\tilde{u}(x) = 1 + \delta^2 x^2, \rightarrow C = 2\delta^2. \qquad (3)$$

We set a spatial-dependent diffusion term $\alpha(x)$ with a small deviation from a fixed value,

$$\alpha(x) = \alpha_0(1 + \varepsilon x^2), \qquad (4)$$

with $\alpha_0, \varepsilon \in \mathbb{R}^+$ and $\varepsilon \ll 1$. If $\varepsilon = 0$, $\alpha(x) = \alpha_0$ for the whole domain.

Our toy model is useful in exercising all the discretized operators and necessary skills to tackle the challenges the reduction of a problem alike presents.

1.1 Boundary and Initial Conditions

The boundary conditions are obtained by evaluating the function $u_e(x,t)$ at the domain endpoints, namely

$$b_0(t) = u_e(0,t) = \tilde{f}(t)\tilde{u}(0) = \tilde{f}(t), \qquad (5a)$$
$$b_L(t) = u_e(L,t) = \tilde{f}(t)\tilde{u}(L) = \tilde{f}(t)\left(1 + \delta^2 L^2(t)\right). \quad (5b)$$

We get the initial condition from an evaluation of $u_e(x,t)$ at time zero, namely

$$u_0(x) := u_e(x,0) = \tilde{f}(0)\tilde{u}(x). \qquad (6)$$

1.2 ALE Ingredients

There are two kind of moving meshes: one that keeps fixed the domain length and simply moves the interior nodes; and one that moves the mesh according to a scaling law that changes the domain length.

1.2.1 Fixed Length

An adaptive mesh technique would want to move the mesh nodes according to some feature of the solution or its gradients. In this case, despite having a fixed domain, $L = L_0$, the interior nodes would move, leading to an ALE formulation. The mapping and mesh velocity would be

$$\mathcal{A}(\mathcal{X},t) = \mathcal{X} - k\sin\left(\frac{2\pi\mathcal{X}}{L_0}\right)\sin(\omega t), \qquad (7a)$$

$$w(\mathcal{X},t) = -\omega k\sin\left(\frac{2\pi\mathcal{X}}{L_0}\right)\cos(\omega t) \qquad (7b)$$

The drawback of this mapping is that we need to solve a non-linear equation,

$$(\mathcal{X} - x) - k\sin(2\pi\mathcal{X})\sin\left(\frac{2\pi t}{T}\right) = 0, \qquad (8)$$

to obtain $\mathcal{X} = \mathcal{X}(x,t)$ to compute the mesh velocity in physical coordinates, $w(x,t)$, which is the one that needs to be provided to the weak form in the actual calculation of the solution at the algebraic level.

1.2.2 Changing Length

The alternative scenario is one were the domain length is not fixed, $L = L(t)$. The movement of the rightmost side $L(t)$ is defined with a harmonic function,

$$L(t) = L_0\left[1 - \sin(\omega t)\right], \qquad (9a)$$
$$L'(t) = -L_0\omega\cos(\omega t). \qquad (9b)$$

where the frequency ω will be our geometrical parameter. The mapping and mesh velocity are

$$\mathcal{A}(\mathcal{X},t) = \left(\frac{\mathcal{X}}{L_0}\right)L(t), \qquad (10a)$$

$$w(\mathcal{X},t) = \left(\frac{\mathcal{X}}{L_0}\right)L'(t), \qquad (10b)$$

$$w(x,t) = \left(\frac{1}{L_0}\right)L'(t)L_0\left(\frac{x}{L(t)}\right) = L'(t)\left(\frac{x}{L(t)}\right) \quad (10c)$$

In this scaled moving domain it is simpler to compute the mesh velocity in physical coordinates, since we do not need to solve a non-linear equation to obtain $\mathcal{X} = \mathcal{X}(x,t)$.

We could decide to stop the simulation when the domain has stretched by a factor n. If so, we have to compute the final time t_f,

$$L(t_f) = nL_0, \quad 0 < n < 1, \qquad (11)$$

so that the end time t_f of our simulation should be

$$t_f = \frac{1}{\omega}\sin^{-1}(1 - n). \qquad (12)$$

1.3 Forcing Term

The strong formulation forcing term $f(x,t)$ from which we depart is computed by introducing $u_e(x,t)$ into the PDE, which leads to

$$f(x,t) = \tilde{f}'(t)\tilde{u}(x) - \alpha(x)\tilde{f}(t)\tilde{u}''(x), \qquad (13a)$$

$$f(x,t) = \tilde{f}'(t)\tilde{u}(x) - 2\alpha(x)\delta^2\tilde{f}(t). \qquad (13b)$$

Since we did not obtain $u_e(x,t)$ from any reasoning related to the body of the PDE, the forcing term is not identically zero. This was precisely our goal, to obtain a forcing term which would lead the solution of the PDE to be $u_e(x,t)$.

1.3.1 Forcing Due to Lifting

We recall that the FE problem we solve is the one corresponding to the homogeneous solution of the problem.

Therefore, in order to implement these problems we need to define the exact expression of f_g. From Equation (**??**) we get that the forcing term linked to the lifting of the Dirichlet boundary conditions has the following form,

$$f_{g,1} = -\frac{\partial g}{\partial t} + w\frac{\partial g}{\partial x}, \tag{14}$$

$$f_{g,2} = -\alpha\frac{\partial g}{\partial x}. \tag{15}$$

Since the lifting g is the linear combination of both boundary conditions, Equation (**??**),

$$g(x,t) = b_L(t)\left(\frac{x}{L}\right) + b_0(t)\left(\frac{L-x}{L}\right), \tag{16}$$

we get that the forcing term f_g becomes the combination of

$$\frac{\partial g}{\partial x} = \frac{b_L(t) - b_0(t)}{L}, \tag{17a}$$

$$\frac{\partial g}{\partial t} = b_L'(t)\left(\frac{x}{L}\right) + b_0'(t)\left(\frac{L-x}{L}\right). \tag{17b}$$

The time derivatives of the boundary conditions are derived from Equations (5),

$$b_0'(t) = \tilde{f}'(t), \tag{18a}$$

$$b_L'(t) = \tilde{f}'(t)\left(1 + \delta^2 L^2(t)\right) + 2\tilde{f}(t)\delta^2 L(t)L'(t). \tag{18b}$$

The expression for $\tilde{f}'(t)$ will be different for each manufactured problem. Note that the second summand of $b_L'(t)$ vanishes if $L(t)$ is constant, i.e. $L'(t) = 0$.

1.4 MFP-1: Reach a Steady-State Solution from Zero

Our first problem will be built such that a zero initial condition is given, and that a steady state solution is reached.

For $\beta \in \mathbb{R}^+$, let the function $\tilde{f}(t)$ be

$$\tilde{f}(t) = 1 - e^{-\beta t}, \tag{19a}$$

$$\tilde{f}'(t) = \beta e^{-\beta t}. \tag{19b}$$

The shape of the time function $\tilde{f}(t)$,

$$\begin{cases} \tilde{f}(0) = 0, \\ \lim_{t\to\infty}\tilde{f}(t) = 1, \end{cases}$$

ensures that we reach a steady state solution for large time values,

$$\lim_{t\to\infty} u_e(x,t) = \tilde{u}(x), \tag{20}$$

and that we have a null initial condition.

If we define the number f_∞ to be smaller than unity, but very close to it, $f_\infty \sim 1$, we can estimate the final simulation time to arrive to the steady-state solution by solving for t in Equation (19a),

$$t_f \simeq -\frac{1}{\beta}\ln(1 - f_\infty). \tag{21}$$

The larger β, the quicker the steady-state arrives, but also the smaller the discrete timestep Δt will have to be to capture correctly the time evolution of the system.

The explicit expression for the forcing term is obtained from Equations (13) and (4),

$$\begin{aligned} f(x,t) &= \beta e^{-\beta t}(1 + \delta^2 x^2) \\ &\quad - 2\delta^2\alpha_0(1 + \varepsilon x^2)\left(1 - e^{-\beta t}\right). \end{aligned} \tag{22}$$

The boundary condition values, to be used to build the lifting term in Equation (**??**), are obtained from Equations (5),

$$b_0(t) = 1 - e^{-\beta t}, \tag{23a}$$

$$b_L(t) = \left(1 - e^{-\beta t}\right)\left(1 + \delta^2 L^2(t)\right). \tag{23b}$$

Collecting all the parameters used to define problem MFP-1, we find:

- α_0: constant diffusion constant.
- ε: present in the diffusion coefficient, it determines how sensitive $\alpha(x)$ is to spatial variations.
- ω: Present in the definition of the moving domain, it defines how quickly the domain boundary moves.
- δ: present in the moving boundary condition and in the forcing term. It regulates how strongly the solution changes in the spatial direction.
- β: Present in both boundary conditions and in the forcing term, it defines how quickly the steady-state is reached.

So, $\mu = [\alpha_0, \varepsilon, \omega, \delta, \beta]$ and $|\mu| = 5$.

1.5 MFP-2: Unsteady for All Times

Our second problem will be built such that we depart from an initial profile and that a steady state is never reached.

For $\omega_f \in \mathbb{R}^+$, let the function $\tilde{f}(t)$ be

$$\tilde{f}(t) = \cos\left(\omega_f t\right), \tag{24a}$$

$$\tilde{f}'(t) = -\omega_f\sin\left(\omega_f t\right). \tag{24b}$$

The shape of the time function $\tilde{f}(t)$,

$$\begin{cases} \tilde{f}(0) = 1, \\ \left\|\tilde{f}(t)\right\| < \infty, \\ \nexists\lim_{t\to\infty}\tilde{f}(t), \end{cases}$$

ensures that we start from an initial profile at the beginning of the simulation,

$$u_e(x,0) = \tilde{u}(x), \tag{25}$$

and that we never reach a steady-state solution due to the oscillatory nature of $\tilde{f}$.

Again, the explicit expression for the forcing term is obtained from Equations (13) and (4),

$$\begin{aligned} f(x,t) &= -\omega_f\sin\left(\omega_f t\right)(1 + \delta^2 x^2) \\ &\quad - \delta^2\alpha_0(1 + \varepsilon x^2)\cos\left(\omega_f t\right). \end{aligned} \tag{26}$$

The boundary condition values, to be used to build the lifting term in Equation (**??**), are obtained from Equations (5),

$$b_0(t) = \cos\left(\omega_f t\right), \tag{27a}$$

$$b_L(t) = \cos\left(\omega_f t\right)\left(1 + \delta^2 L^2(t)\right). \tag{27b}$$

Collecting all the parameters used to define problem MFP-2, we find:

- α_0: constant diffusion constant.
- ε: present in the diffusion coefficient, it determines how sensitive $\alpha(x)$ is to spatial variations.
- ω: Present in the definition of the moving domain, it defines how quickly the domain boundary moves.
- δ: present in the moving boundary condition and in the forcing term. It regulates how strongly the solution changes in the spatial direction.
- ω_f: Present in both boundary conditions and in the forcing term, it defines how quickly the solution changes in time.

So, $\mu = [\alpha_0, \varepsilon, \omega, \delta, \omega_f]$ and $|\mu| = 5$.