Chaotic mixing in a rectangular cavity: theoretical background

Martien A. Hulsen & Patrick D. Anderson

January 7, 2009

1 Introduction

Numerous reasons exist to be interested in mixing. Convective heat transfer by mixing in fluids is, for example, of interest in thermal engineering where flows are used to achieve a uniform temperature distribution. In astrophysics and combustion, mixing through turbulence is studied. Understanding of viscous and viscoelastic fluid mixing is of importance in material and food processing, and is our main application area in this project. In most of these cases the goal is to control the stirring process: to enhance or suppress mixing. The significance of mixing in food preparation is obvious for every cook. The importance of polymer mixing is illustrated by the fact that only a few percent of all new polymers are ever commercialized. Due to the high cost of synthesizing new materials, the much more inexpensive alternative is mixing (blending) of existing polymers to obtain a final product with the desired properties.

Although in industry mixing of materials is carried out in static, batch, and dynamic mixers, which can have complex geometries, theoretical analyses so far mainly report on much simpler systems. Well-known examples of industrial mixers are single (see Fig. 1) or twin-screw extruders. The example we will be studying is the rectangular cavity flow and is of direct importance for these types of mixers. We will be analyzing the mixing properties of this flow by

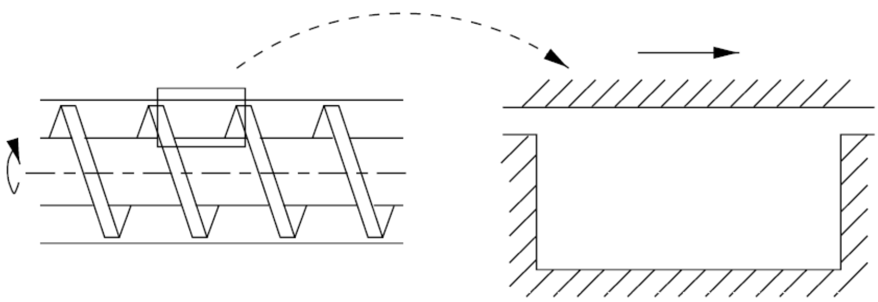

Figure 1: On the left a part of a single-screw extruder is plotted. Because of the geometrical complexities of single and twin-screw extruders, prototypical systems like cavities are studied. This figure shows how they are related.

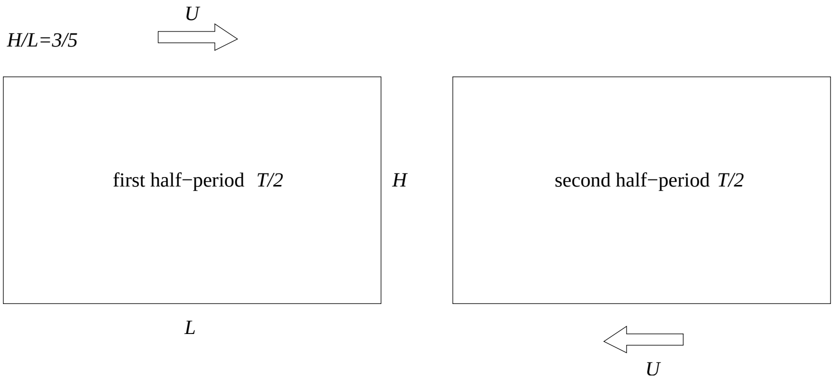

Figure 2: A time-periodic flow in a rectangular cavity: during the first half period $T/2$ the upper lid is moved to the right with a velocity U and during the second half period $T/2$ the lower lid is moved to the left with a velocity of also U.

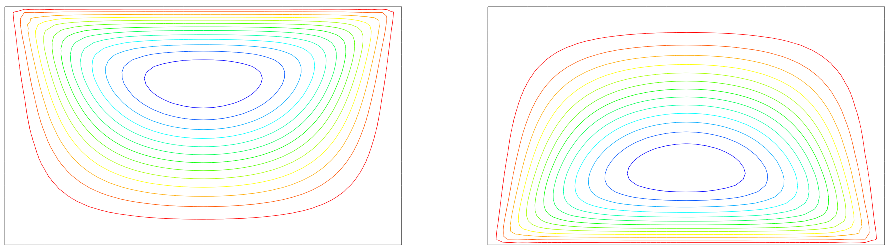

Figure 3: Left: streamlines in first half period; Right: streamlines in the second half period.

tracking in time a blob of material that is passively convected by the flow. In particular we will be studying the increase of the length of the interface between the material of the blob and the surrounding material. In case of good mixing, this length should grow exponentially.

2 Model problem: a time-periodic flow in a rectangular cavity

It is very well known that a steady incompressible flow in 2D cannot lead to chaos and hence to effective mixing[1]. Therefore, moving the upper lid of a 2D cavity as suggested in Fig. 1 with a constant velocity will not be an effective mixing protocol. However, a time-periodic mixing protocol as shown in Fig. 2 can lead to chaotic mixing. For a Newtonian flow where inertia terms can be neglected (Stokes flow) the streamlines have been plotted in Fig. 3. The mixing efficiency will depend on the ratio of the height H and length L of the cavity

[1]A standard reference book on mixing is the book by Ottino [1].

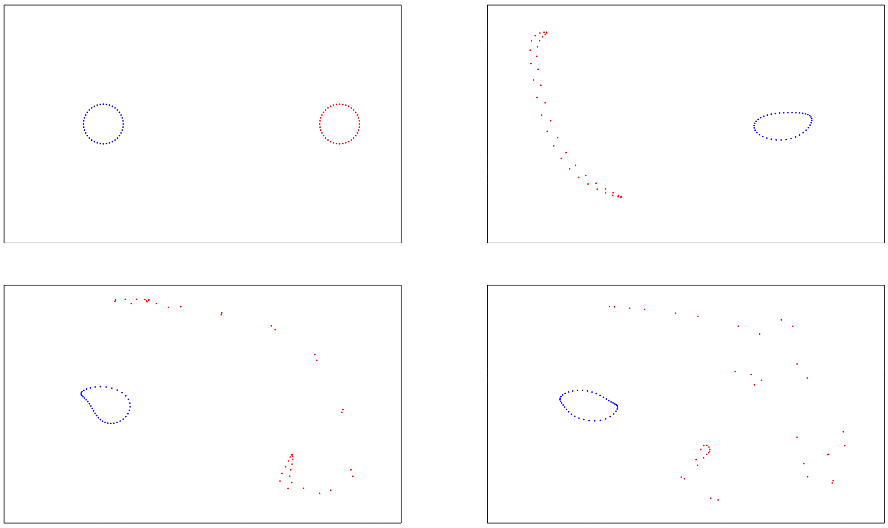

Figure 4: The positions of two blobs at times $t = 0$, $T/2$, T and $2T$. $D = 6.24$

and the dimensionless displacement

$$D = \frac{UT}{2L} \tag{1}$$

where T is the period (one upper lid movement and one lower lid movement). In this project we will use $H/L = 3/5$, which leaves D as the only parameter.

When looking at the actual mixing of material blobs we will notice that the mixing efficiency depends on the position of the blob. For example, in Fig. 4 two blobs are tracked that start from a different position in the region. The blobs are represented by 40 points on the interface between the material in the blobs and the outside fluid. After tracking the points it is clear that the material of the blue (left) blob remains close together, whereas the red (right) blob is highly stretched (and folded, which is more difficult to see). In fact, the length of the interface hardly changes for the blue blob whereas the length of the interface growth exponentially for the red blob, which is the blueprint of effective mixing[2].

Looking at Fig. 4, it is clear we have a numerical problem when tracking exponentially growing interfaces with a constant number of points. We need an adaptive interface tracking scheme that can increase the number of points where needed. This will be the main topic of this project.

[2]The two blobs are positioned around so-called periodic points, i.e. points that return to their original position after one period. The blue blob is positioned around an 'elliptic' point where material only rotates and the red blob is positioned around a 'hyperbolic' point where the material is highly stretched. Finding and analyzing periodic points is one of the main methods to analyze mixing in time-periodic flows. Another method is the so-called Poincaré map. Both methods are limited to time-periodic flows and are beyond the scope of this project. The tracking of interfaces, which we use in this project, can be applied to any flow.

3 Particle tracking

For tracking the individual points at an interface we need to solve the following ordinary differential equation for the position vector $\boldsymbol{x}$:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}(\boldsymbol{x}, t), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2}$$

where $\boldsymbol{x}_0$ is the initial position of the particle and $\boldsymbol{u}(\boldsymbol{x}, t)$ is the velocity field. The field $\boldsymbol{u}(\boldsymbol{x}, t)$ must be known, either in an analytical form or obtained from a numerical program that solves the flow problem. In this project we will be using an analytical form:

first half-period (left picture in Fig. 2):

$$\psi(x, y) = -\frac{UH}{2\pi} \sin(\pi \frac{x}{L}) \sin\left[\pi(\frac{y}{H})^2\right] \tag{3}$$

second half-period (right picture in Fig. 2):

$$\psi(x, y) = -\frac{UH}{2\pi} \sin(\pi \frac{x}{L}) \sin\left[\pi(1 - \frac{y}{H})^2\right] \tag{4}$$

where ψ is the stream function and $\boldsymbol{x} = (x, y) = (0, 0)$ is the lower-left corner of the domain. The velocity field $\boldsymbol{u} = (u, v)$ can be found by differentiating:

first half-period:

$$\begin{aligned} u(x, y) &= \frac{\partial \psi}{\partial y} = -\frac{Uy}{H} \sin(\pi \frac{x}{L}) \cos\left[\pi(\frac{y}{H})^2\right] \\ v(x, y) &= -\frac{\partial \psi}{\partial x} = \frac{UH}{2L} \cos(\pi \frac{x}{L}) \sin\left[\pi(\frac{y}{H})^2\right] \end{aligned} \tag{5}$$

second half-period

$$\begin{aligned} u(x, y) &= \frac{\partial \psi}{\partial y} = U(1 - \frac{y}{H}) \sin(\pi \frac{x}{L}) \cos\left[\pi(1 - \frac{y}{H})^2\right] \\ v(x, y) &= -\frac{\partial \psi}{\partial x} = \frac{UH}{2L} \cos(\pi \frac{x}{L}) \sin\left[\pi(1 - \frac{y}{H})^2\right] \end{aligned} \tag{6}$$

Equation (2), together with Eqs. (5) and (6) will be solved by a second-order Runge-Kutta time-integration scheme:

$$\begin{aligned} \boldsymbol{x}^*_{n+1} &= \boldsymbol{x}_n + \Delta t \boldsymbol{u}(\boldsymbol{x}_n, t_n) \\ \boldsymbol{x}_{n+1} &= \frac{1}{2} \Delta t \big[\boldsymbol{u}(\boldsymbol{x}^*_{n+1}, t_{n+1}) + \boldsymbol{u}(\boldsymbol{x}_n, t_n)\big] \end{aligned} \tag{7}$$

where $\Delta t = t_{n+1} - t_n$ is the time step.

4 Adaptive interface tracking

Before devising the algorithm for adding points on the interface we need to lay down the 'structure' of how we describe the interface. We use a description as used in the finite-element method. The points describing the interface are called the *nodes* and the line connecting the nodes are the *elements*. In Fig. 5 we have plotted a closed interface with eight nodes and eight elements.

First we need to specify the coordinates of the nodes by an array of rank two:

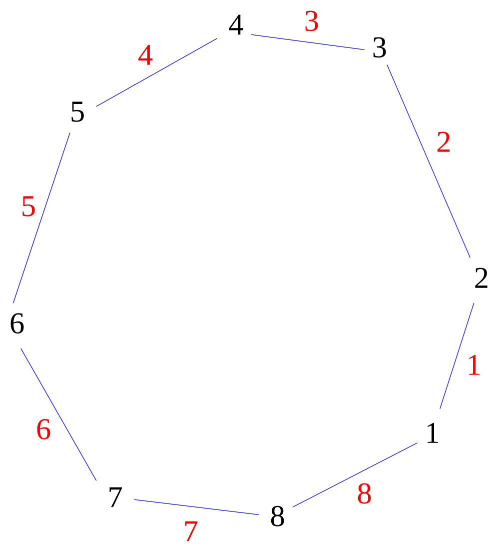

Figure 5: Description of an interface with nodes and elements

```
coor(1:nnodes,1:2)
```

where `nnodes` is the number of nodes, for example `coor(8,1)` is the x-coordinate of node number 8. The topology of the interface will be specified by an array of rank two:

```
topology(1:nelem,1:2)
```

where `nelem` is the number of elements, for example `topology(5,1:2)` will give the two node numbers element 5 is connected to. For the interface in Fig. 5 we have

$$\texttt{topology} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 6 \\ 6 & 7 \\ 7 & 8 \\ 8 & 1 \end{pmatrix} \tag{8}$$

We will also need an array which gives for each nodal point the two elements that are connected to that point. For the interface in Fig. 5 we have for that array

$$\texttt{nodelem} = \begin{pmatrix} 8 & 1 \\ 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 6 \\ 6 & 7 \\ 7 & 8 \end{pmatrix} \tag{9}$$

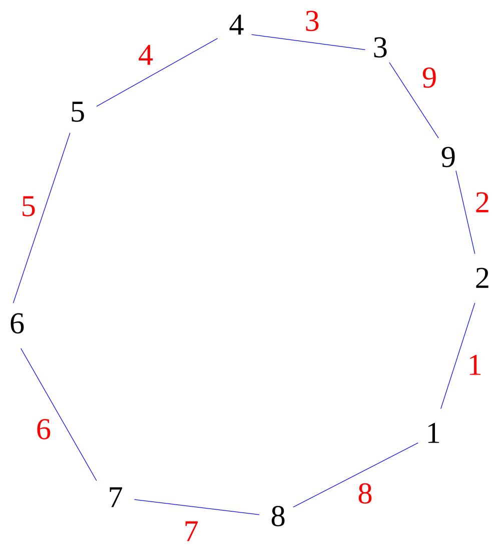

Figure 6: Adding a node (and an element) to an interface

The structure is not static, since in the adaptive scheme we need the possibility to add nodes. For example, assume that some accuracy criterion (see below) is exceeded in element number 2 in Fig. 5. We want to split this element into two elements by adding a node and an element. This has been done in Fig. 6. The addition of a node and an element increases the first dimension of the arrays `coor`, `topology` and `nodelem` by one. The arrays `topology` and `nodelem` now become

$$\texttt{topology} = \begin{pmatrix} 1 & 2 \\ 2 & 9 \\ 3 & 4 \\ 4 & 5 \\ 5 & 6 \\ 6 & 7 \\ 7 & 8 \\ 8 & 1 \\ 9 & 3 \end{pmatrix} \qquad \texttt{nodelem} = \begin{pmatrix} 8 & 1 \\ 1 & 2 \\ 9 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 6 \\ 6 & 7 \\ 7 & 8 \\ 2 & 9 \end{pmatrix} \tag{10}$$

The accuracy criterion for adding nodes we will be using is that

1. the length of an element h, i.e. the distance between two consecutive nodes, must be smaller than $h_{\max}$.

2. in strongly curved parts of the interface the length of the elements must be smaller than h_{cmax}. Strongly curved means that the (absolute value of the) angle between two consecutive elements is larger than $\alpha_{\max}$ (see Fig. 7).

There is one question still unanswered: what should the position of a newly added node be? The simplest procedure would be to just put the point halfway

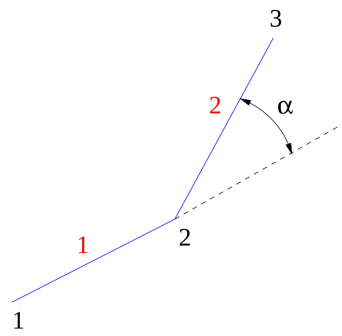

Figure 7: The angle α between two consecutive elements

between the two already existing two adjacent nodes (nodes 2 and 3 in Figs. 5 and 6). However this usually leads to relatively large errors. Therefore we choose to start from a situation where the two adjacent nodes are still sufficiently close. Assume we divide each half period T into a number of substeps (`nsteps`) and check the accuracy criterion for all nodes/elements at each substep and split the elements that do not satisfy the criterion. We now interpolate between the position of the two adjacent nodes *at the previous step* and track the new point to the current step. This means, we need to store the coordinates of the nodes at a previous time by an array of rank two:

```
coorprev(1:nnodes,1:2)
```

where now `nnodes` is the number of nodes at the previous time.

Once we have an interface defined by the adaptive procedure above we have to compute the length of the interface. For this we simply take the length of the polygon defining the interface, that is the sum of the lengths of the straight lines between the nodes.

The flows we will be considering are assumed to be incompressible. This means that, in theory, the area of the blob must remain constant. An easy way of computing the area A by a boundary integral is given by

$$A = \frac{1}{2}\int_{\Omega} \nabla \cdot \boldsymbol{x}\, d\Omega = \frac{1}{2}\oint_{\Gamma} \boldsymbol{n} \cdot \boldsymbol{x}\, d\Gamma \tag{11}$$

where Ω is the inner region of the blob, Γ is the interface and $\boldsymbol{n}$ is the outside normal. Since each element is straight, the integral can be performed analytically on each element. If we denote the first and second nodal point of an element by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively, the difference vector by $\boldsymbol{v} = \boldsymbol{x}_2 - \boldsymbol{x}_1$ and the sum vector by $\boldsymbol{s} = \boldsymbol{x}_1 + \boldsymbol{x}_2$, then the area is given by (the absolute value of)

$$A = \frac{1}{4}\sum_{k}^{N_e}(v_x^k s_y^k - v_y^k s_x^k) \tag{12}$$

where N_e is the number of elements. Note, that A is positive for a clockwise nodal numbering and negative for counterclockwise. The preservation of the area as a function of time can be used to monitor the quality of the interface tracking.

5 Conclusion

In this paper we have described a numerical tool to analyze the mixing efficiency of 2D flow in different parts of the flow. The analysis consists of positioning blobs of material in different parts of the flow and monitoring the growth of the length of the interface as a function of time. The growth should be exponential in order to achieve efficient mixing and the rate is a measure for the efficiency.

Problems

1. Argue that parameter D in Eq. (1) and the ratio H/L fully determines the mixing behavior of the time-periodic cavity flow.

2. Explain why for the time-periodic cavity flow the explicit time-dependence in the right-hand size of Eq. (2) is needed.

3. Summarize what variables the 'structure' that describes the interface must contain and of what type they are (scalar, array, dimension and size of the array, integer or real, ...).

4. Is it really necessary to keep the array `nodelem`? Isn't it redundant?

5. Describe the algorithm for adding a node to the structure in a general way suitable for programming. How would it change for tracking multiple blobs? What about a non-closed surface?

6. The arrays in the structure change their size when adding nodes. How would that best be programmed? Would you use arrays that are reallocated such that they exactly match the number of nodes? Why not?

7. Show that for points on a circle we have $h \approx R\alpha$ and use this to relate the curvature of the interface to the value of α.

8. Derive an expression for α (as defined in Fig. 7) for given coordinates $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ that can be readily programmed.

9. We have formulated a criterion for adding new nodes on page 6. If the conditions are violated in a particular element we split the element into two elements by adding a node. Are we always sure the conditions are fulfilled after the split? Do we need more iterations to fulfill the conditions?

10. In the main text on page 6 we describe the process of adding a node, i.e. starting from an interpolated position at a previous step. Another possibility would be to always start from the situation at the beginning of a half period. How would that change the algorithm? A further possibility is to always start from the situation at the beginning (i.e. the original blob). This would make the algorithm much more complicated. Why?

11. Derive an expression for the length ℓ of the interface, similar to the expression for the area A in Eq. (12).

12. In theory the area of a blob should remain constant in an incompressible flow. In practical computations this is only approximately fulfilled. What are the sources of the error?

13. Verify Eqs. (11) and (12).

14. Consider a circular interface described by n uniformly distributed points. The approximation of the interface by piecewise straight lines in this case is called a regular polygon. Show that the *relative* error in the length ℓ is given by

$$E_{r\ell} = \frac{\theta - 2\sin\frac{\theta}{2}}{\theta} \approx \frac{1}{24}\theta^2 = \frac{1}{6}(\frac{\pi}{n})^2 \tag{13}$$

and the *relative* error in the area A is given by

$$E_{rA} = \frac{\theta - \sin\theta}{\theta} \approx \frac{1}{6}\theta^2 = \frac{2}{3}(\frac{\pi}{n})^2 \tag{14}$$

where $\theta = 2\pi/n$, the angle of a single segment of the polygon. The approximation is valid for small θ.

References

[1] J.M. Ottino. *The kinematics of mixing: stretching, chaos and transport.* Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, 1989.