# Real Time Visualization of Flow Mixing on GPU

**Alejandro Aguilar-Sierra**[1] **and Luis Miguel de la Cruz-Salas**[2]
[1] Centro de Ciencias de la Atmósfera, [2]Instituto de Geofísica
Universidad Nacional Autónoma de México
asierra@unam.mx

## 1. Abstract

We study a mixing flow produced by a natural convection process. This flow is generated by imposing time-dependent oscillating temperatures on two opposite walls of a square where a Newtonian and incompressible fluid is contained. The flow consists of a vortex moving itself around the center of the square, that can be interpreted as a combination of a translating rotating mixer with blinking vortices. These conditions generate chaotic mixing flows where no moving walls are required. The visual analysis of the flow is done using a Lagrangian tracking of particles [4] in combination with the diffusive strip method [5]. The latter technique, requires too much time to compute the concentration diffusion of a contaminant. In [1] we described the implementation of this technique in CUDA. In this work we also implement the particle tracking in CUDA along with and strip refining method applying cubic splines. These new implementations allow us to start our visual analysis with few particles, and we can add new particles as needed in order to avoid the emergence of visual artifacts. The particle position and the resulted diffused scalar field is drawed as an OpenGL texture and displayed in real-time.

## 2. Time-dependent natural convection setup

Consider a two dimensional square filled with a Newtonian and incompressible fluid. The temperatures of the left half of the upper wall, and the right half of the bottom wall, are cyclically changed as shown in figure 1.

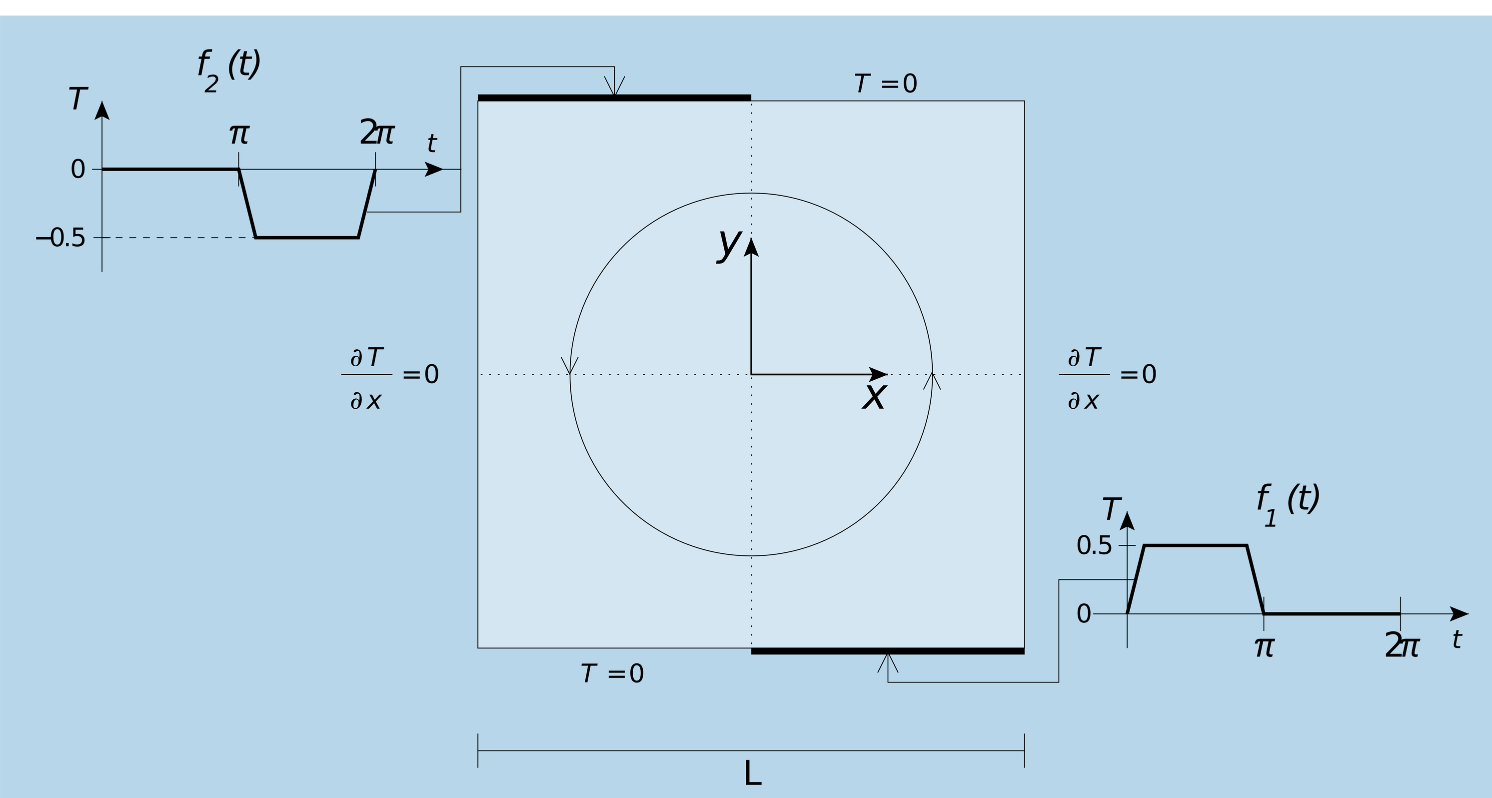


**Figure 1:** *Square cavity geometry and boundary conditions.*

The governing equations in terms of non-dimensional variables are (see [3]):

| Mass | Navier-Stokes | Energy |
|---|---|---|
| $\nabla\cdot\mathbf{u}=0$ | $\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}=-\nabla p+\nabla\cdot(\mathrm{Pr}\nabla\mathbf{u})+\mathbf{g}$ | $\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T=\mathrm{Pr}\nabla^2 T$ |

where $\mathbf{u}, p$ and $T$ are respectively the velocity, the pressure and the temperature of the fluid. These equations were solved using the software TUNAM [2] which is a C++ library that implements the control volume method.

## 3. Advective and Diffusive Mixing

The mixing efficiency of the flow can be qualitatively assessed from the simultaneous Lagrangian tracking of a set of $N$ passive tracers in the flow by integrating the equation of motion

$$\frac{d\mathbf{x}_i}{dt}=\mathbf{u}(\mathbf{x}_i,t),\quad \forall i=1,\ldots,N \tag{1}$$

where $\mathbf{x}_i$ is the position of the tracer $i$ at instant $t$, which has the velocity $\mathbf{u}(\mathbf{x}_i,t)$ given by the numerical solution of the governing equations. In this work we use the fourth order Runge-Kutta method to integrate the equation (1).
Our objective is to follow the position of a passive contaminant, distributed along a thin strip, and calculate its diffusion as it flows. The approach used to determine the markers concentration $c$, is based on the Diffusive Strip Method [5], where final equation for $c$ on a given position $\mathbf{x}$ of the domain at the instant $k$ is given by

$$c^k(\mathbf{x})=\sum_{i=1}^{N}\frac{c_0/1.7726}{\sqrt{1+4\tau_i^k}}\exp\left(-\frac{[(\mathbf{x}-\mathbf{x}_i^k)\hat{\sigma}_i^k]^2}{\Delta l^2}-\frac{[(\mathbf{x}-\mathbf{x}_i^k)\hat{n}_i^k]^2}{(s_i^k)^2(1+4\tau_i^k)}\right) \tag{2}$$

where $\hat{\sigma}_i^k$ and $\hat{n}_i^k$ are the unit vectors tangent and normal to the strip, see figure 2. In equation (2), $\mathbf{x}_i^k$ represents the position of the particle $i$ of the strip, at time iteration $k$, and $\mathbf{x}$ represents positions taken on a very fine mesh of the domain. The process consists in summing the contribution of all Gaussian ellipses defined by equation (2) in each point of this fine mesh.

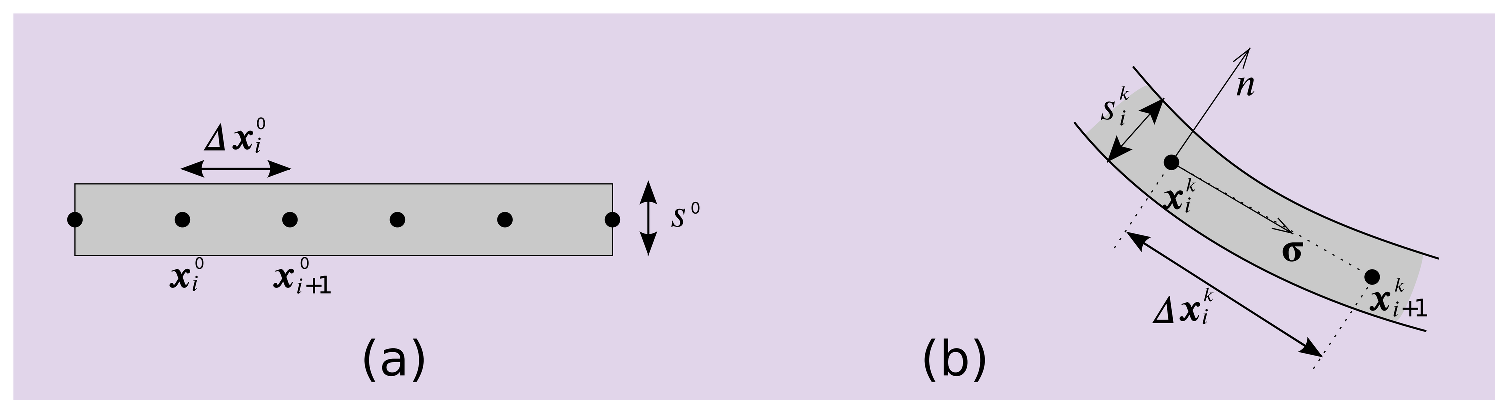


**Figure 2:** *(a) A schematic strip defined by the set of tracers $\mathbf{x}_i$ for $i=1,\ldots,N$ with a striation thickness $s^0$. (b) A section of the strip deformed by the flow after $k$ time steps.*

## 4. Mixing flow visualization on the GPU

The Lagrangian particle tracking and the concentration calculation, consisting respectively of the integration of equation (1) and the evaluation of formula (2), exhibit rich amount of data parallelism suitable to be done in the GPU.

### 4.1 Concentration calculation

The evaluation of formula (2) must be done in a very fine mesh of the domain. This process, for several cycles of the heating-cooling protocol as shown in figure 1, take long time. The calculation of this process in the GPU is straightforward. In order to represent a 2D mesh of dimension $512\times512$ we executed a grid of 16 by 16 blocks, each block of 32 by 32 threads. Each thread computes the formula (2) for a single point in the grid. This thread configuration gives us a high-performance balance between the number of threads per block, the maximum number of blocks per multiprocessor in the chip, and the available resources.

### 4.2 Lagrangian particle tracking

The tracking of the particles requires more effort. First, we need to move the information of the flow to the GPU. We only need to send to the GPU memory, the velocity field in the neighborhood of the strip. With that information available, it is possible to do all the tracking on the GPU. Additionally, it is required to add or delete points on the strip depending on the curvature of the strip in each time step. To do this, in each step, we update the position of the current points, and these will be used as the control points to compute a cubic spline curve. Using these curves, we can populate the curve with points along the splines separated according to the local curvature.

## 5. Real-time visualization

With the concentration field in the GPU memory, we take advantage of CUDA and OpenGL interoperability to visualize the results in real-time. First we create an OpenGL Pixel Buffer Object and allocate memory space for a texture with the same dimension as the grid we used to compute the concentration. The CUDA kernel creates an image representing the concentration as gray levels, zero represented as white, to be used as an OpenGL texture.

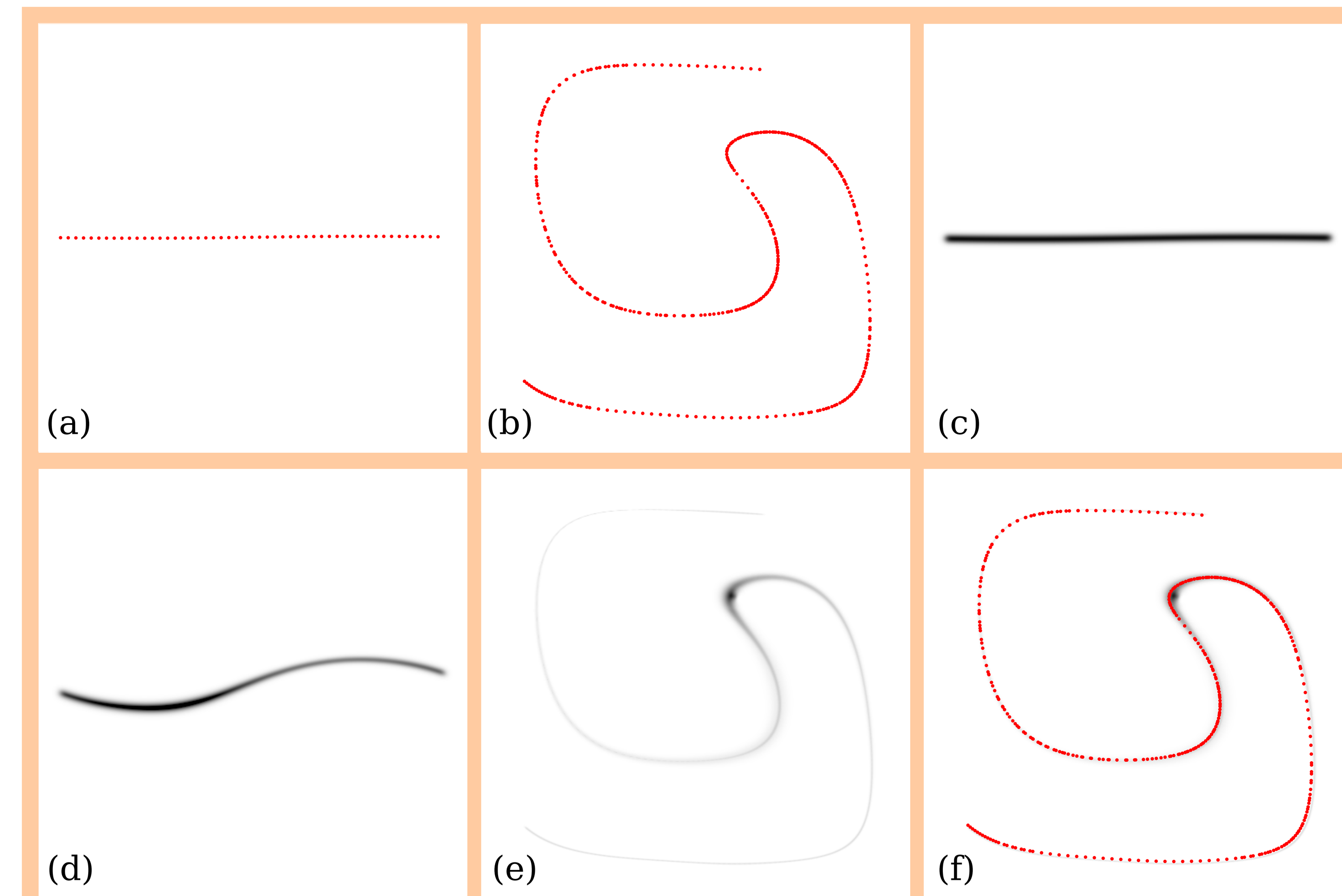


**Figure 3:** *a) Initial number of points on the strip; b) Points on the strip after 0.5 cycles; c) Initial concentration; d) Concentration after 0.2 cylces; e) Concentration after 0.5 cycles; f) Concentration and points after 0.5 cycles.*

## Conclusion

We have shown that using a simple CUDA kernel and OpenGL interoperability, it is possible to study the problem of mixing in cavities for several cycles in real-time. Particularly, in this problem we were able to analyze 16 cycles in less time than the required for 2 cycles in the CPU version without simultaneous visualization.

## References

[1] Aguilar-Sierra A. and L. M. de la Cruz Salas. Accelerated flow visualization of advective–diffusive mixing processes using gpus. *Acta Universitaria*, 22(5), 2012.

[2] L. M. de la Cruz. Tunam: Templates units for numerical applications and modeling. Homepage: http://code.google.com/p/tunam/.

[3] L.M. de la Cruz and E. Ramos. Mixing with time dependent natural convection. *International Communications in Heat and Mass Transfer*, 33(2):191 – 198, 2006.

[4] Luis M. de la Cruz, Ian Garcia, Victor Godoy, and Eduardo Ramos. Case study: parallel lagrangian visualization applied to natural convective flows. In David E. Breen, Alan Heirich, and Anton H. J. Koning, editors, *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, pages 41–44. IEEE, 2001.

[5] P. Meunier and E. Villermaux. The diffusive strip method for scalar mixing in two dimensions. *J. Fluid Mech.*, 662:134–172, 2010.