Electronic Journal: Southwest Journal of Pure and Applied Mathematics
Internet: http://rattler.cameron.edu/swjpam.html
ISSN 1083-0464
Vol. No. 01, 1997, pp. 1–7.
Submitted: November 15, 1996. Published: April 29, 1997.

NUMERICAL STUDY OF FLAME PROPAGATION IN BOUNDARY LAYERS

Maria E. Calzada

Abstract. An incompressible, reactive, viscous flow reaches a hot circular cylinder causing a reaction to start. We observe the progress of the reaction, concentrating in the transition towards turbulence within the boundary layer. The problem is divided into the fluid dynamics effects and the combustion effects. The motion of the flow is resolved by the use of a hybrid vortex algorithm; the reactive part of the problem is modeled by a combination of a temperature equation and a flame advection and propagation algorithm that mimics Huyghens' principle. The following paper will explain these methods, how to implement them, and will present the numerical results of the experiment.

(1991) A.M.S. Subject Classification Codes. 76D05, 76M25.

Key Words and Phrases. Fluid Dynamics, Reactive Flows, Navier-Stokes Equations, Vortex Methods.

1. HYDRODYNAMICS EFFECTS

The motion of an incompressible, viscous flow is governed by the Navier-Stokes equations, which in vorticity transport form can be written as:

$$\begin{gathered} \partial_t\xi + (\vec{u}\cdot\vec{\nabla})\xi = Re^{-1}\Delta\xi \\ div(\vec{u}) = 0, \end{gathered} \tag{1}$$

where $\vec{u} = (u, v)$ is the velocity field, $\xi = curl(\vec{u}) = \partial_x v - \partial_y u$, Re is the Reynolds number associated with the flow, and Δ is the Laplacian operator. To observe turbulent effects we will choose the Reynolds number to be large.

We will solve the Navier-Stokes equations in an indirect way, by using a stream function ψ and a hybrid vortex algorithm. The stream function ψ has the properties that $u = \partial_y\psi$, $v = -\partial_x\psi$, and therefore $\Delta\psi = -\xi$. The idea of the vortex blobs method is to solve for ψ, given ξ, and then by differentiating ψ obtain $\vec{u} = (u, v)$.

Near the boundary of the cylinder, the Navier-Stokes equations can be approximated by Prandlt's boundary layer equations, which in vorticity transport form can be written as:

$$\begin{aligned} \partial_t\xi+(\vec{u}\cdot\vec{\nabla})\xi &= \nu\partial_y^2\xi \\ div(\vec{u}) &= 0 \\ \xi &= -\partial_y u. \end{aligned} \tag{2}$$

Mathematics and Computer Science Department, Loyola University New Orleans
New Orleans, Louisiana, 70118, U.S.A.
E-mail Address: calzada@beta.loyno.edu
©1997 Cameron University

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

These equations are only valid in a very thin region near the boundary of the cylinder due to the turbulent effects. In this region we will use the vortex sheets method to resolve the motion of the flow.

The vortex methods were originally developed by Alexander Chorin in the 1970's [1, 2]. They are Lagrangian methods best suited for flows at large Reynolds numbers. The vortex methods have been used successfully in many applications [3, 4, 5, 6, 7]. We will first describe the vortex blobs method and then the vortex sheets methods.

The vortex blobs method: The basis of the vortex blobs method is to solve an inviscid flow with no boundaries. Such a flow is regulated by Euler's equations in terms of vorticity transport in the entire plane:

$$\begin{aligned} \partial_t \xi + (\vec{u} \cdot \vec{\nabla})\xi &= 0 \\ div(\vec{u}) &= 0. \end{aligned} \tag{3}$$

Suppose we have a collection of N vortex blobs, approximations of point vortices, with circulations $\xi_1, \xi_2, \ldots, \xi_N$, located at the points $\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N$ respectively, where $\vec{r} = (x, y)$. Assume there are no other contributions to the vorticity field. We can write the vorticity field in the following way:

$$\xi(\vec{r}) = \sum_{j=1}^{N} \xi_j \xi_o(\vec{r} - \vec{r}_j), \tag{4}$$

where ξ_o is the following smooth approximation to the Dirac delta function:

$$\xi_o(\vec{r}) = \begin{cases} \frac{1}{2\pi\sigma\|\vec{r}\|}, & \|\vec{r}\| < \sigma, \\ 0, & \|\vec{r}\| \geq \sigma. \end{cases} \tag{5}$$

σ is a cut-off to be determined. We can use this equation to find an appropriate stream function ψ, and by differentiation we can find the velocity field $\vec{u}_\xi$ that regulates the motion of the vortex blobs and the rest of the flow. Using this velocity field we are able to reproduce the successive motion of the blobs; for example, if we use Euler's method to move the blobs, their position at the next time step will be given by:

$$\vec{r_i}^{n+1} = \vec{r_i}^{n} + \Delta t \vec{u}_\xi(\vec{r_i}^{n}), \tag{6}$$

where $\vec{r_i}^{n} = \vec{r}_i(n\Delta t)$, and Δt is the time step.

We now make a correction to this flow by introducing boundaries. This creates the boundary condition $\vec{u}\cdot\vec{n} = 0$ ($\vec{n}$ is a unit vector normal to the boundary). We satisfy this boundary condition using a potential flow $\vec{u}_p = (u_p, v_p)$, solving the following problem: find φ so that $\partial_x \varphi = u_p$ and $\partial_y \varphi = v_p$ satisfying $\Delta\varphi = 0$, in D and $\partial_n \varphi = -\vec{u}_\xi \cdot \vec{n}$ on ∂D, where D is the computational domain, and ∂D is the boundary of D. As can be verified, this potential flow would cancel any normal component of velocity at the boundary of the cylinder introduced by the vortex blobs without affecting the vorticity of the flow. The position of the blobs is now approximated by:

$$\vec{r_i}^{n+1} = \vec{r_i}^{n} + \Delta t[\vec{u}_\xi(\vec{r_i}^{n}) + \vec{u}_p(\vec{r_i}^{n})]. \tag{7}$$

We make a second correction to this flow by finally introducing viscosity, that is $Re^{-1} \neq 0$; however, for the moment we disregard boundaries. The equations governing the flow are the full Navier-Stokes equations (1). We simulate this viscosity by adding a random walk to the blobs. Let $\vec{\eta_i}^{n} = (\eta^n_{x_i}, \eta^n_{y_i})$, where $\eta^n_{x_i}$ and $\eta^n_{y_i}$ are normally distributed random variables with mean 0 and variance $2\Delta t/Re$. The motion of the blobs is now regulated by:

$$\vec{r_i}^{n+1} = \vec{r_i}^{n} + \Delta t[\vec{u}_\xi(\vec{r_i}^{n}) + \vec{u}_p(\vec{r_i}^{n})] + \vec{\eta_i}^{n}. \tag{8}$$

In the pure vortex blobs method we would consider the boundary at this point and would deal with the no-slip condition, $\vec{u} \cdot \vec{s} = 0$, caused by the viscosity ($\vec{s}$ is a unit tangent vector to the boundary). To cancel this tangential velocity component at the wall, vorticity would be created. However, in our hybrid algorithm, we utilize the Prandlt's boundary layer equations and the vortex sheets method to create vorticity at the boundary. We discuss this method next.

The vortex sheets method: The original vortex sheets method was designed using a flat plate as the boundary of the flow [2]. However, a simple change of variables allows us to use virtually the same equations when the boundary of the flow is a circular cylinder (see figure 1). All calculations for the sheets take place in the xy-plane while their real location is in the x_1y_1-plane, with relations $\theta = \frac{x}{2\pi r}$, $x_1 = (r+y)\cos\theta$, and $y_1 = (r+y)\sin\theta$. It is clear that the computational domain for the xy-plane needs to be periodical in the x direction with period $2\pi r$, i.e., when $x > 2\pi r$, x needs to be replaced by $x - 2\pi r$ to have $0 \leq x \leq 2\pi r$. Integrating Prandlt's boundary layer equations (2) and solving appropriately for u and v in a way that automatically satisfies the boundary condition $\vec{u}\cdot\vec{n} = 0$, we can obtain the following equations for the velocity field:

$$u(x,y) = U_\infty(x) - \int_y^\infty \xi dy. \tag{9}$$

$$v(x,y) = -\int_0^y \partial_x u dy. \tag{10}$$

In these equations $U_\infty(x)$ is the free stream velocity. Within the boundary layer, vorticity is carried in sheets. We seek to simulate this effect accurately by introducing vortex sheets S_i. They are parallel to the boundary, have length h and are centered at $r_i = (x_i, y_i)$. The intensity of a vortex sheet is defined to be $\xi_i = -(\vec{u}\cdot\vec{s}^{\,+} - \vec{u}\cdot\vec{s}^{\,-})$, that is the jump of the tangential velocity across the sheet. We choose notation similar to that of vortex blobs because that will ease the transition and coupling between the sheets and the blobs. In a way much like the motion of vortex blobs, we can advance the position of the vortex sheets using (9) and (10):

$$x_i^{n+1} = x_i^n + \Delta t u_i \tag{11}$$

$$y_i^{n+1} = y_i^n + \Delta t v_i + \eta_i, \tag{12}$$

where η_i's, drawn from a normal distribution with mean 0 and variance $2\nu\Delta t$. The random walk is only applied in the y direction because in the Prandlt's equations diffusion is neglected in the x direction.

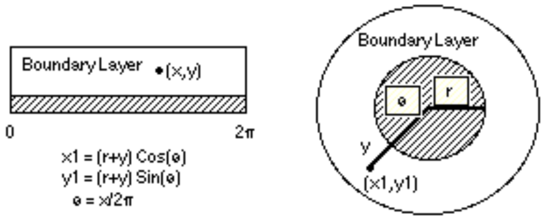

Figure 1. Illustration of Change of Variables for Sheets

To satisfy $\vec{u}\cdot\vec{s}$, we create sheet elements at the boundary in the following way: we take m segments of length h parallel to the boundary. We compute the tangential component of the velocity field at the centers of these segments, and imagine a vortex sheet of strength twice this velocity. We create sheets to cancel this velocity, incorporate the newly created sheets into the flow by a random walk, and let the sheets participate in the subsequent motion of the fluid.

Coupling the sheets and the blobs: Our hybrid vortex algorithm uses sheets near the boundary, where Prandlt's equations are valid, and blobs away from the boundary, where Prandlt's equations break down and the full Navier-Stokes equations regulate the flow. The boundary layer is proportional to $1/\sqrt{Re}$. From boundary layer theory we know that near the separation point the velocity gradient $(\vec{u}\cdot\vec{s})/(\vec{u}\cdot\vec{n})$ is very large. We use this criteria to assign a transition between sheets and blobs: Sheets having velocity gradient $(\vec{u} \cdot \vec{s})/(\vec{u} \cdot \vec{n})$ greater than a certain chosen parameter are close to the separation point and thus are turned into blobs. This parameter is determined by the geometry of the obstacle and by the mean value of the velocity gradients of all sheets. A sheet S_i with intensity ξ_i that leaves the boundary layer becomes a blob with circulation $h\xi_i$. As determined in [8] the cut-off σ is taken to be h/π. To ensure the consistency of the velocity across the boundary layer, imagine a sheet sitting on the edge of the boundary layer. It could be treated as a blob, and thus would be affected by all the other blobs. So all sheets should be affected by all blobs as well. But blobs are only affected by other blobs.

2. THE COMBUSTION SCHEME

In addition to the Navier-Stokes equations (1), the energy of the system is regulated by the following temperature equation:

$$\partial_t T + (\vec{u} \cdot \vec{\nabla})T = \Delta T + \Omega, \tag{13}$$

where Ω is a reaction term, and which is coupled to the Navier-Stokes equations through the velocity field $\vec{u}$. Equation (13) can be solved in a number of ways. We solved it by using operator splitting in the following way:

$$\partial_t T + (\vec{u} \cdot \vec{\nabla})T = 0 \tag{14}$$

$$\partial_t T = \Delta T \tag{15}$$

$$\partial_t T = \Omega \tag{16}$$

Equation (14) was solved using the random choice method presented in [6]; equation (15) was solved using Crank Nickolson [9]; and equation (16) was dealt with in the coupling of the SLIC algorithm with a flame dictionary to be discussed next.

SLIC and a flame dictionary: [10] discusses the construction of a flame dictionary that could be used to catalog the flame speed and preheat thickness of a given reaction. This dictionary can be coupled with different front tracking algorithms that would simulate the motion of the flame front. The current work couples the flame dictionary with a flame advection and propagation algorithm that simulates Huyghens' principle. Other front tracking algorithms like [11] can be used in conjunction with a flame dictionary.

SLIC (Simple Line Interface Calculation) was developed by Noh and Woodward [12] and has been successfully used to model a number of combustion problems. SLIC reconstructs the shape of the flame front using vertical and horizontal lines and moves it in a direction normal to itself using an appropriate velocity. Let the fluid occupy a domain D, with boundary ∂D, in two dimensions. The fluid subdomain $D_1(t) \subseteq D$ is burnt; the fluid in subdomain $D_2(t) = D - D_1(t)$ is unburnt. $D_1(t)$ is expanding; $D_2(t)$ is contracting. The boundary between $D_1(t)$ and $D_2(t)$ is an interface between the two fluids. This is the burning front that is both advected by the hydrodynamic velocity of the fluid and also moves with a speed $U(T)$, the flame speed, in a direction normal to itself. Superimpose a grid of mesh h on D, and consider a matrix $0 \leq f_{i,j}(n) \leq 1$ that at time n accounts for the proportion of burnt fluid contained in cell (i, j); we call this the matrix of burnt fractions. When called, SLIC reconstructs the shape of the burning front in each cell and moves it given the appropriate component of velocity. For more information on how SLIC reconstructs the flame front see [6].

The flame front is advected with the motion of the fluid. To mimic this SLIC reconstructs the front in sweeps. In a vertical sweep SLIC uses the vertical component of the velocity field from the vortex methods to move the flame front. In a horizontal sweep, SLIC uses the horizontal component of a velocity field from the vortex methods to move the flame front. At the end of the sweeps, $f_{i,j}(n)$ is updated to account for the new burnt fractions.

The flame front is also advanced within a direction normal to itself with a speed that depends on the chemical composition and on temperature. We use a flame dictionary of temperatures to obtain the appropriate flame speed for a given cell. We model the motion of the flame front in the direction normal to itself using the following numerical implementation of the Huyghens' principle: Consider eight angles equidistributed over the unit circle $\theta_l = l\pi/4, l = 1, \cdots, 8$. Move the flame front with the appropriate flame speed in each or those eight directions. Let $f_{ij}^{\theta_l}(n)$ be the new matrix of burnt fractions obtained by moving the flame front in each of the eight directions θ_l. Let $f_{ij}^{\theta_o}(n) = f_{ij}(n)$. We approximate the burnt region at time $n+1$ by

$$f_{ij}(n+1) = \max_{0 \le l \le 8} f_{ij}^{\theta_l}(n). \tag{17}$$

Note that under this numerical implementation of Huyghens principle having more than 8 equidistributed angles over the unit circle are unnecessary. Each cell (i,j) has exactly 8 immediate neighbors. The amount of mass transported from cell (i,j) to any of its neighbors is largest if the translation vector points from the center of cell (i,j) to the center of the neighboring cell. Those directions are the ones we are using in our implementation. Any other direction would not affect $f_{ij}(n+1)$.

We take care of equation (16) at this moment. The unburnt temperature of the fluid is given by T_u. When the fluid is burns, this temperature rises to $T_b = 1+T_u$. We mimic this by updating the temperatures of the cells that were newly burnt to the appropriate burnt temperature $1+T_u$ that related to their unburnt temperature. Care must be taken that the right unburnt temperature is used or the motion of the flame will be artificially accelerated.

3. NUMERICAL RESULTS OF EXPERIMENT

Figure 2 depicts the streamlines of a developing velocity field at different times, for $Re = 2000$. We can observe large vortices behind the circle and small vortices near the boundary of the cylinder due to the vorticity creation to satisfy the no-slip condition.

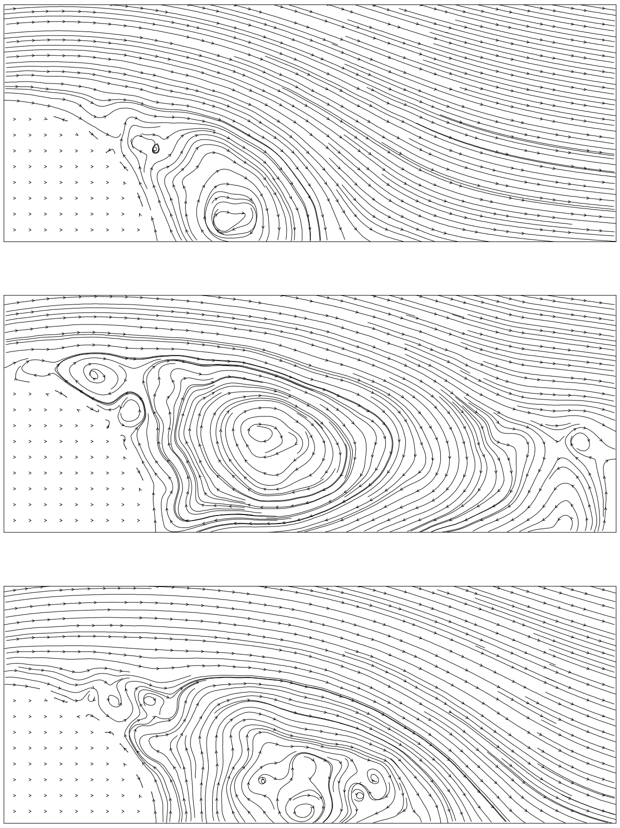

Figure 2. Typical Stream Lines at Different Time Steps

We ran the hybrid vortex procedures on a hot circular cylinder until we encountered well defined recirculation areas as the ones shown in Figure 2. We assumed that until that point the flow was not reactive. The fluid temperature was initially set at .05. The cylinder was initially set at .4. This differential in temperature caused a thermal boundary layer. The temperature diffused and was advected with the flow. After the flow developed nicely, a reactive fluid reached the cylinder and a spark ignited the fluid at the left edge of the cylinder. The reaction propagated as a flame front and moved with the flow and with the flame speed corresponding to the different temperatures.

Figure 3 represent temperature contour plots near the small recirculation areas of the circle were separation occurs. We observe jagged flame fronts due to the recirculation of the flow. We also observe how the recirculation both speeds up the burning process, due to more fluid being accessible to the front, and slows down the burning process, due to the burnt fluid being sent back onto itself.

The numerical simulation has very good qualitative agreement with experimental observations [13]. Other numerical methods agree with the current simulation [7]. The method described in [7] works very well; however it requires a large amount of memory to keep a dictionary of boundary value problems. This may become too expensive in certain settings. The current method uses a small dictionary of flame speeds; however it can prove to be slower than [7] in some cases. The quantitative modeling of the stochastic combustion parameters are left for a future study.

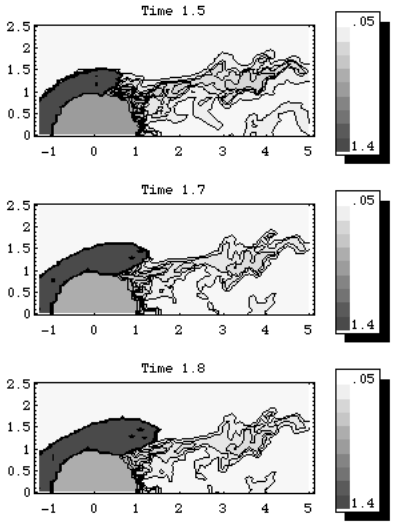

Figure 3. Temperature Profiles at Different Time Steps

Acknowledgements

The author would like to thank Dr. Yu Song for the many helpful discussions and comments he provided.

References

1. Chorin, A.J., *Numerical study of slightly viscous flow*, J. Fluid Mech. **57** (1973), 785-796.
2. Chorin, A.J., *Vortex sheet approximation of boundary layers*, J. Comput. Phys. **22** (1978), 428-442.
3. Ghoniem, A.F., Chorin, A.F., and Oppenheim, A.K., *Numerical modelling of turbulent flow in a combustion tunnel*, Philosophical Transactions of the Royal Society of London **A 304** (1982), 303-325.
4. Sethian, J. A., *Vortex methods and turbulent combustion*, Lectures in Appl. Math. **22** (1985), 245-259.
5. Sethian, J.A. and Ghoniem, A.F., *A validation study of vortex methods*, J. Comput. Phys. **74** (1988), 283-317.
6. Calzada, M.E. and Song, Y., *A numerical algorithm for flame advection and propagation*, Mathematics and Computers in Simulations **35** (1993), 79-92.
7. Song, Y. and Calzada, M.E., *A dictionary random choice method for turbulent combustion*, J. of Comput. Appl. Math. **58** (1995), 27-42.

8. Cheer, A.Y., *Numerical study of incompressible slightly viscous flow past blunt bodies and airfoils*, SIAM J. Sci. Statist. Comput. **4** (1983), 685-705.
9. Sod, G.A., *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws*, J. Comp. Phys. **27 No. 1** (1978), 1-31.
10. Calzada, M.E. and Song, Y., *Application of a flame dictionary method in turbulent premixed flame modeling*, Appl. Math. Letters (to appear).
11. Glimm, J., Isaacson, E., Marchesin, D., and McBryan, O., *Front tracking for hyperbolic systems*, Adv. Appl. Math. **2** (1981), 91-119.
12. Noh, W. and Woodward, P., *A simple line interface algorithm*, Lecture Notes in Physics. Fifth Internat. Conf. on Fluid Dynamics (A. I. van de Vooren and P. J. Zandbergen, eds.), Springer, New York, 1976, pp. 330-340.
13. Lewis, B. and von Elbe, G., *Combustion, Flames and Explosions of Gases*, 3rd ed., Academic Press, New York, 1987.