# A Foul Weather Fountain

Ernie Esser, Ryan Card, Jeff Giansiracusa

February 24, 2002

**Abstract**

The task is to devise a fountain control algorithm to monitor wind conditions and ensure that a fountain at the center of a plaza fires water high enough to be dazzling, while not drenching the pedestrian areas surrounding the fountain pool. We begin by constructing a model of a water fountain based on physical principles of falling water droplets applied to a particle system. Using this model we can examine the behavior of a fountain under various wind conditions through computer simulation. Using complex analytic techniques, we model the wind flow through the plaza and obtain an estimate of how anemometer readings from a nearby roof-top relate to the plaza conditions. We construct four algorithms–two *intelligent* algorithms, a *conservative* approach, and an *enthusiastic* system–to control the fountain. We devise a measure of unacceptable spray levels outside the fountain and use this criterion to compare performance. First we examine the behavior of these algorithms under a set of general abstracted wind conditions. Then we construct a wind signal generator which simulates the conditions of several major cities from meteorological database data [2], and we compare the performance of our control systems in each city. Next we extend our model to examine several interesting possibilities, including tilting the fountain nozzle into the wind, fountains with multiple nozzles, and fountains with non-circular basins. Simulations show that the Conservative and Enthusiastic algorithms both perform unacceptably in realistic conditions. The Weighted Average Algorithm works best in gusty cities such as Chicago, but the Averaging Algorithm is superior in calmer cities such as L.A. and Seattle.

# Contents

# 1 Introduction

A large fountain is installed as a monument in the center of an open plaza in a city. People stop to eat lunch, enjoy the day, and possibly gaze at the fountain. It would be a shame if the fountain were to do anything less than blast water gloriously high up into the air. However, soon after being installed a windy autumn day comes along, and the spray from the fountain drenches crowds of people. To avoid this in the future an anemometer, used to measure wind speed and direction, will be installed on the roof of a nearby building. It will monitor wind conditions, and a control algorithm will adjust the height of the fountain to a safe level minute by minute.

• **The control algorithm cannot possibly respond to changes in conditions at anything below the 10 second scale,** since wind is highly variable and the response of the anemometer is somewhat slow [4]. The goal is therefore to design the algorithm to operate on a time scale of 10 seconds up to a couple hours and adapt the height of the fountain to a maximum safe level.

# 2 Outline of Our Approach

- We begin by constructing a computer model of a water fountain based on physical principles of falling water droplets applied to a particle system.
- Using this model we can examine the behavior of a fountain under various wind conditions. Our control algorithm knows only the anemometer readings and does not have direct knowledge of the conditions at the location of the fountain.
- We model the wind flow through the plaza to obtain an estimate of how the anemometer readings relate to the plaza conditions.
- We construct four algorithms–two *intelligent* algorithms, a *conservative* approach, and an *enthusiastic* system–to control the fountain.
- We devise a measure of unacceptable spray levels outside the fountain, and use this criterion to compare performance.
- First we examine the behavior of these algorithms under a set of general abstracted wind conditions.
- Then we construct a wind signal generator which simulates the conditions of several major cities from meteorological database data [2] and compare the performance of our control systems at each.
- Next we extend our model to examine several interesting possibilities, including tilting the fountain nozzle into the wind, fountains with multiple nozzles, and fountains with non-circular basins.

# 3 Constructing a Model of the Water Jet

We choose to model the spray from the fountain as a particle system. As water droplets spew forth from the nozzle, they are subjected to forces (gravity, air drag, turbulence, etc.). We formulate a simplified differential equation governing the motion and then numerically integrate to find the trajectory for each droplet. This equation will be based on a physically realistic model of small droplets (around 1 mm radius), and will scale up to an effective model for larger clumps of water (up to 10 cm across) because the physics of turbulence and viscosity at this scale cannot possibly be computed accurately.

We will need the following assumptions:

- The drag force is proportional to the square of velocity and the square of radius [12].
- Droplets will break apart into smaller droplets when subjected to wind. Breakup rate is proportional to relative wind speed and surface area [9].
- When a droplet breaks, turbulence will cause the new droplet fragments to move slightly away from their initial trajectory.

## 3.1 Modeling a Single Droplet

The above considerations lead us to formulate the motion of a water droplet as:

$$m\frac{d\vec{v}}{dt} = -mg\hat{z} - \eta|w|^2\hat{w}r^2 \tag{1}$$

where $\vec{v}$ is the velocity vector, $\vec{w}$ is the wind velocity vector relative to the motion of the droplet (wind vector minus velocity vector), $m$ and $r$ are the droplet's mass and radius respectively, and $\eta$ is a constant of proportionality. According to Ref.[11], a raindrop with radius of 1 mm falls at a terminal velocity of 7 m/s. Given this, it is easy to determine the value of $\eta$, resulting in $\eta = 0.855$ kg/m$^3$. Note that larger drops will fall quickly, and very tiny drops will fall very slowly, mimicking a fine mist that hangs in the air for a long time. This is precisely the effect that we wish to capture.

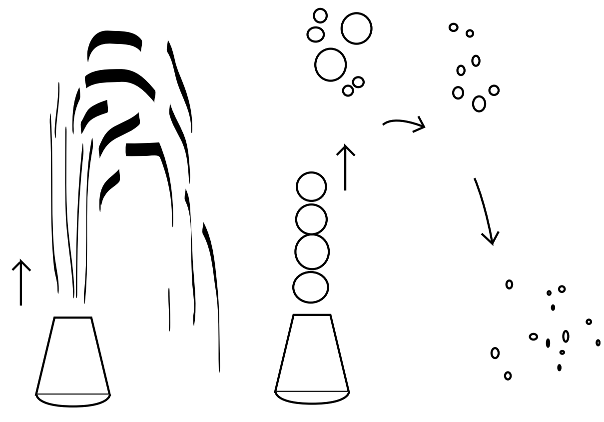
In our model, we assume droplet breakup is a modified Poisson process, with rate $\lambda_{breakup}$ given by,

$$\lambda_{breakup} = \lambda_0|w|r^2. \tag{2}$$

If the breakup rate did not depend on variable parameters $|w|$ and $r^2$, the process would be a standard Poisson process. We will determine $\lambda_0$ by fitting the water stream of our fountain to the streams of two real fountains–the Jet D'eau of Geneva, Switzerland, and the Five Rivers Fountain of Lights in Miami, Florida.

When a breakup event occurs, we split the droplet into two new droplets and divide the mass of the parent in a random proportion (from a uniform distribution). Air turbulence will tend to impart to the two new droplets a small velocity component perpendicular to the relative wind direction $\vec{w}$. This

Figure 1: A continuous water jet is approximated by a discrete stream of water blobs.




is the effect which causes a tight stream of water to spread out as it travels, even under zero-wind conditions. We choose this velocity nudge to have magnitude 2% of the particle's velocity relative to the air, and in a random perpendicular direction. We give the two drops resulting from a breakup equal and opposite nudges.

## 3.2 Putting Water Drops Together to Make a Fountain

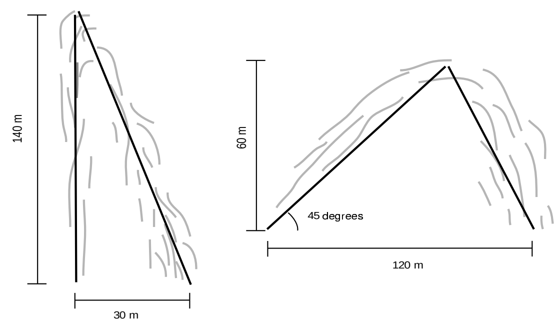
To construct a water fountain from the single water drop model above, we define the water jet as a stream of large water drops. Their size is roughly the size of the nozzle, and they leave with an initial velocity equal to the nozzle's output velocity. This is shown schematically in figure 1. The water blobs leave at a rate such that the flux of water is equal to the flux given by a nozzle-sized cylindrical stream moving at the same speed.

To model the turbulence in the jet as the water leaves the nozzle, we give each water blob a normal distribution of radius and initial velocity.

- The standard deviation of blob radii is 10% of the nozzle size.
- The standard deviation of initial velocities is 5%.
- The blobs leave with an angular spread of 3°. This is consistent with industrial high-pressure nozzles [10].

• **Comment:** A very important effect in projectile streams, such as the one we are modeling here, is that the wind drag is significantly reduced for particles following one another closely (NASCAR drivers and racing cyclists are

Figure 2: The Jet D'Eau and the Fountain of Lights.




intimately familiar with this phenomenon). We have not made any explicit mention of modeling this phenomenon because these effects are already incorporated into the dynamics of large water blobs (which can be thought of as representing many small drops moving together). This is why we consider the equations for large drops to be an effective model rather than a realistic interaction model.

### 3.3 Fitting our Fountain to the Jet D'Eau and the Five Rivers Fountain of Lights

The Five Rivers Fountain of Lights, located in Dayton, Florida, is one of the largest fountains in the world [5]. It consists of several water jets, and on low-wind days each propels a water stream 60 m high and 120 m out. The Jet D'Eau, located in Geneva, Switzerland, is another impressive fountain. It shoots a 30 cm diameter stream of water at 60 m/s straight up. The water reaches a height of 140 m, and on an average breezy day (wind speed 5 m/s on Lake Geneva), the water returns to earth approximately 35 meters downwind from the nozzle [7, 6]. See figure 2 for a schematic.

To analyze the accuracy of our model, and determine a reasonable value for the breakup parameter $\lambda_0$, we first matched our geometry to the Five Rivers Fountain of Lights. We fixed $\lambda_0$ so that with an initial velocity chosen so that the stream reaches a height of 60 m, the stream will return to the ground at a distance of just over 100 m. A value of $\lambda_0$ too large results in the water breaking up too quickly into tiny droplets which have a much lower terminal velocity and thus failing to reach the desired distance. If the value is too small, then an unrealistically small amount of spray is produced and the water blob will travel too far.

The results are summarized in table 1. We set $\lambda_0 = 5000$, and note that the results are highly insensitive to this parameter. Varying $\lambda_0$ by a factor of 2 cause only 15% changes in the distances. Therefore, even though our method for determining this parameter is fairly rough, the important behavior is much more strongly affected by other parameters which have been set more precisely.

Table 1: Comparison between real fountains and our model.

| | Jet D'Eau | | Five Rivers Fountain of Lights | |
|---|---|---|---|---|
| | real | model | real | model |
| Height (m) | 140 | 121 | 60 | 62 |
| Distance (m) | 35 | 30 | 120 | 100 |

We conclude from this comparison that our model reproduces the spray patterns of known fountains accurate to within about 15%. However, these two fountains are both rather extreme cases and we expect that for a plaza-sized fountain our model will be more accurate since our formulae for breakup and drag force are derived under less extreme conditions. Therefore, we trust this model as an accurate testing ground for fountain control algorithms.

# 4 Wind Flow Through the Plaza

Buildings and other structures in an urban environment may cause significant disturbances to wind flow patterns. Rooftop and street-level conditions can often be quite different. Therefore, the data readings from a rooftop anemometer could give a very distorted picture of wind conditions in the plaza below if these effects are not properly accounted for.

To model the plaza wind, we first make the following assumptions:

- There are no significant structures between the buildings at each side of the plaza.
- The plaza is large. Effects caused when wind flow leaves the plaza are negligible at the plaza-center. The significant effects are entirely caused at the inward boundary passage.
- The air flow is smooth enough so that turbulent vortices are negligible.

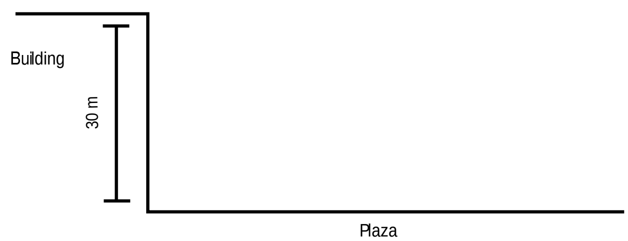
## 4.1 Formulation

Given the above assumptions, we may approximate the geometry of the plaza as in figure 3, and make use of complex analytic flow techniques ([1] p. 225). With a Schwarz-Cristoffel mapping of a smooth horizontal flow from the upper half of the complex plane onto the region above the plaza, we obtain a flow function for the wind as it enters the plaza area. The flow function for this region is given by,

$$\Gamma_c(t) = \frac{h_0}{\pi}\{[(t+ic)^2-1]^{1/2} + \log(t+ic+[(t+ic)^2-1]^{1/2})\}, \tag{3}$$

where $t$ parametrizes a streamline for each value of $c$. These streamlines are plotted in figure 4 where the acceleration of the wind as it passes over the building edge and the decreased velocity in the plaza are both clearly visible.

Figure 3: Schematic representation of the relevant features of the plaza




The flow velocity $\vec{v}$ is inversely proportional to the streamline spacing, so the horizontal component of it is calculated as follows,

$$v_x = \mathrm{Im}[\frac{\partial \Gamma_c}{\partial c}]. \tag{4}$$

The horizontal velocity profile for a streamline which passes about three meters above the building roof (corresponding to $c = 0.6$) is plotted in figure 4. We chose this streamline because three meters is a reasonable height for an anemometer mounting. From these graphs one can see that the wind speed through the plaza center (at a distance of 30 to 40 m from edge) is *approximately half* of the rooftop wind speed. This calculation is validified by its excellent agreement with the findings of Santamouris (Ref. [3]), who reports that in flows perpendicular to a street, the ground level velocities are between zero and 55% of the free stream velocities.

## 4.2 Results

We take the following conclusions from this flow model:

- **Placement of the anemometer is important!** It should be mounted near the center of the rooftop to minimize disturbances from the roof's edge.

Figure 4: Streamlines for wind flow entering the plaza. Decreased wind speed at the plaza level is apparent. Also, note the highly increased wind speed near the edge of the building.

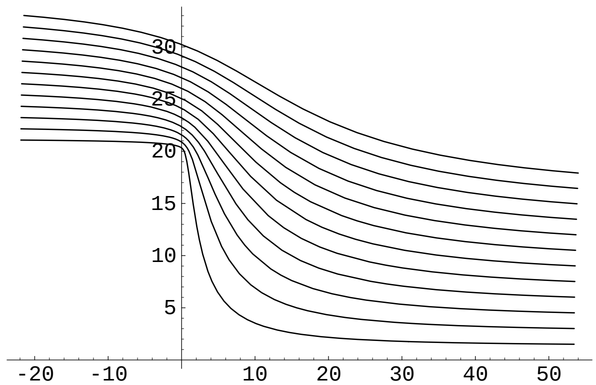
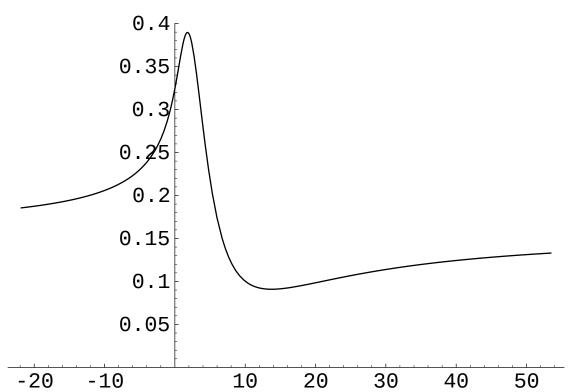



Figure 5: Horizontal velocity profile for the streamline corresponding to $c = 0.6$). This streamline passes above the building's roof at a height of 3 meters, which is a reasonable anemometer mounting height.



- **The anemometer reports a wind speed that is highly biased!** The ground level wind moves approximately half as swiftly as the roof-level wind. Fortunately, it is easy to correct for this bias.
- **Wind speeds are fairly independent of height.** If the fountain is not significantly higher that the surrounding buildings, then it is safe to assume a that the wind is spatially constant.

# 5 Modeling the Wind Variation Over Time

**The control system must be able to handle a range of weather conditions**, from calm up to strongly gusty. We begin by abstracting the wind patterns into three generalized types of increasing complexity.

Type 1: **A low intensity constant breeze of a few m/s.** The model for this is trivial. This type is meant to test the algorithms ability to judge the proper height for a given wind speed.

Type 2: **A breeze varying smoothly over a timescale of a couple minutes.** We use a sinusoidal oscillation in magnitude and direction, with a constant term to reflect the prevailing wind direction of the hour. This type is designed to test the algorithms' ability to adapt to slowly changing conditions.

Type 3: **Sudden unexpected wind gusts, with a few seconds duration and very high intensity.** We model the occurrence of a gust as a Poisson process and distribute the gust durations and intensities normally. The mean and variance are chosen to produce reasonable results. This is perhaps the most important test since it will evaluate the algorithms' caution level. The gusty scenario can easily fool a naive algorithm.

## 5.1 Generating a Realistic Wind Signal

To test the effectiveness of our control systems in a less contrived environment, we would like to construct a model of the wind speed variation over time. Each geographic location may have a very different wind speed profile, so we parametrize the wind profile of a given location by four numbers:

- The **Mean Steady Wind** $\mu_{steady}$ measures the average wind level.
- The **Mean Gust Strength** $\mu_{gust}$ is a measure of an average gust strength, where a gust is defined to be variation on the sub 15 second timescale.
- The **Gust duration** $t_{gust}$ is a characterization of how long the average gust lasts.
- And the **gust deviance** $\sigma_{gust}$ measures the standard deviation of the gust strength distribution.

Table 2: Characteristic parametrization of several major US cities. Note that these parameters specify the plaza wind conditions, which are slightly more mild that the free stream conditions.

| | $\mu_{steady}$ (m/s) | $\mu_{gust}$ (m/s) | $t_{gust}$ (s) | $\sigma_{gust}$ (m/s) |
|---|---|---|---|---|
| Seattle, WA | 1.2 | 2.25 | 6.0 | 0.7 |
| Chicago, IL | 2.0 | 4.0 | 3.0 | 4.0 |
| Boston, MA | 2.3 | 4.2 | 4.0 | 2.2 |
| Los Angeles, CA | 1.7 | 2.0 | 3.0 | 0.7 |
| Washington D.C. | 1.3 | 3.4 | 3.0 | 1.0 |

We construct the realistic wind signal by using these characteristic numbers to combine the abstracted wind types in the following way:

- Type 1 constant wind of strength $\frac{2}{3}\mu_{steady}$,
- Type 2 sinusoidal oscillations of amplitude $\frac{2}{3}\mu_{steady}$,
- and a Type 3 gust signal with mean amplitude of $\mu_{gust}$, amplitude standard deviation of $\sigma_{gust}$, and duration mean $t_{gust}$ and deviation $\frac{1}{2}t_{gust}$.

From WebMET data [2], we estimate these characteristic numbers for a representative sample of major US cities. These numbers are summarized in table 2. Figure 6 shows a comparison of a Seattle wind signal with a Chicago signal. The extreme gustiness of the "Windy City" is apparent.

We also create a *Hurricane Floyd* wind profile by multiplying a Chicago wind signal with a factor which damps the wind to zero early on (the calm before the storm) and then grows to amplify the signal up to Hurricane Floyd level winds over a time period of 10 minutes.

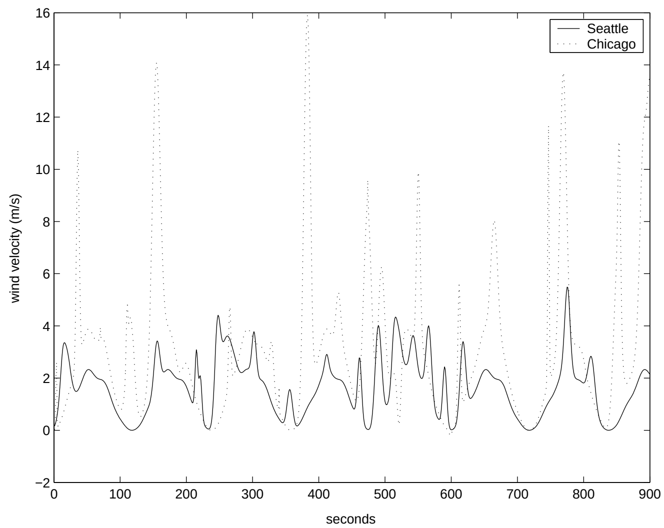
# 6 Development of Fountain Control Algorithms

The goal of the control algorithm is to respond to the anemometer data by maximizing the height fountain while minimizing the probability of the plaza area outside the fountain pool being drenched. The control algorithm will have:

- Access to anemometer readings.
- Direct control over the nozzle velocity.

The control system must have some knowledge of how the water spray travel distance relates to nozzle velocity and wind speed. We wish to tabulate this relationship, but first we must specify our measure of spray distance. We develop two complementary measures: the radius inside of which a given fraction of the water lands, and the farthest distance at which a maximum allowable density of water spray is exceeded. We then tabulate the relationship between nozzle/wind

Figure 6: Horizontal velocity profile.




conditions and spray distance. The algorithms we develop will then combine this table with an estimate of possible future wind speed (based on the current wind and a stored recent history) to make a judgment of a good nozzle velocity.

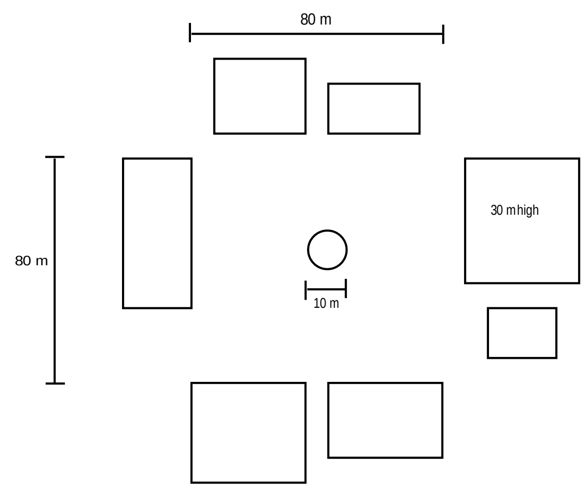
## 6.1 A Measure of Water Spray Distance

- Our first measure is simple: the radius within which 99% of all the sprayed water lands.
- For the second measure, let us assume that the threshold for acceptable water density outside the fountain corresponds roughly to a light rain, which we estimate to be one centimeter in 10 hours ($2.8 \times 10^{-4}$ millimeters of rainfall per second).

**In simulations over a suitably long time period, we find these two measures to agree closely** (to within 1%) and thus any disagreement is a strong indication that a longer simulation time and a larger number of water drops are required.

We will evaluate the performance of our control algorithm by measuring how the spray distance compares to the actual radius of the pool. If the spray radius goes beyond the pool radius, then people in the area might become unacceptably wet. However, if this radius is significantly less than the pool radius, then we are not getting as much height out of the fountain as we possibly could, thus boring the spectators in a way that could be detrimental to local business.

Figure 7: The layout of our hypothetical model plaza.




## 6.2 Constructing the Control System

We begin with a few useful assumptions:

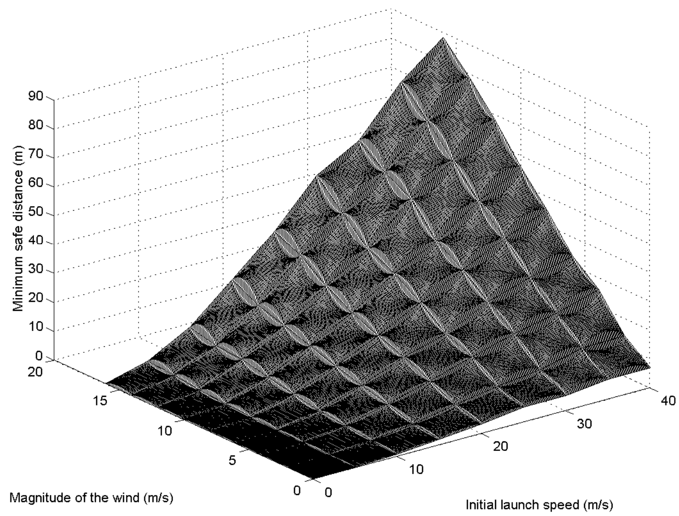
- **Variation in wind direction can be safely ignored by our algorithm** Using the idea of the triangle inequality, if the wind pushes the drop first in one direction and then in another, they will necessarily land nearer to the fountain than if they had been pushed in one direction continuously.
- The algorithm has access to real-time anemometer data averaged over 10-second intervals as well as at least a 10 minute history of wind speed measurements. Even if the anemometer responds faster than 10 seconds, it is nonsensical to try to vary the fountain faster than this because the water requires approximately this much time to complete its flight.
- For concreteness, we will focus on the plaza configuration of figure 7. Most importantly, the fountain is at the center of a circular pool of radius 5 m.

We tabulate the spray distance as a function of wind speed and nozzle velocity (shown in figure 8 ) for use in the algorithm. A quick estimate of how far a water droplet can travel starting at height $z_0$, falling at its terminal velocity $v_t$ and moving at the horizontal wind speed $w$ suggests that

$$distance \approx \frac{z_0 w}{v_t}.$$

The smallest droplets our simulations produce have radii of about 1mm. The corresponding terminal velocity is 7m/s. For specific heights and winds, we find

Figure 8: Linearly interpolated table of spray distance as a function of wind speed and nozzle velocity. Each data point represents a burst of 5 nozzle-size water blobs.




that this rough estimate is usually within 30% of the corresponding minimum safe distance as shown in figure 8 . This is a good indication that our simulations are producing reasonable results.

Naturally, the wind speed is not expected to remain constant over time. An algorithm which optimizes the fountain for the wind at that moment may be caught by surprise if a gust comes. A smart algorithm should therefore consider the history of wind speed measurements when predicting how high the wind speed could get in the near future.

### 6.3 The Control Algorithms

We are now ready to specify the four algorithms which we will simulate and compare.

1. **Averaging Algorithm:** This algorithm considers an average of the previous ten minutes of wind data, and the sample variance. The worst-case scenario is estimated to be a wind strength of one standard deviation above the average.

2. **Weighted Average Algorithm:** The key feature of this algorithm is that when calculating the average and variance of the wind over the last ten minutes, the data are weighted linearly according to recentness. The current measurement gets the highest weight.

3. **Conservative Algorithm:** This algorithm uses the maximum wind speed measured over the last ten minutes to predict what the worst-case wind might be. This is the most conservative approach–it will always err towards safety.

4. **Enthusiastic Algorithm:** This algorithm ignores previous wind data history and puts the fountain to the maximum safe height given the immediate conditions. No precaution is taken with regard to possible future wind behavior.

# 7 Results

## 7.1 Comparing the Algorithms

To compare the effectiveness of our control algorithms, we test each against the following gamut of wind conditions:

- Type 1 Constant wind
- Type 2 smoothly varying wind
- Type 3 Highly variable gusty wind
- Real wind data from Seattle, Chicago, Boston, Los Angeles and Washington D.C.
- Hurricane Floyd type winds!

We run several simulations of the fountain–each for three minutes–under the control of each algorithm. This time interval was chosen to be long enough to capture relevant wind features and give statistical significance to the results. For consistency, we run each algorithm under an identical wind signal (to remove random variation). These three criteria are the basis for comparing the performance of the algorithms:

- The average height the water spout achieves over the time of the simulation.
- The percentage of the total water contained within the confines of the fountain's pool.
- The ratio of the highest density of water landing outside the pool area to the maximum acceptable spray density ($2.8 \times 10^{-4}$ mm/s).

The Weighted Average Algorithm was expected to do the best job of optimizing fountain height, but the results of our simulations, shown in tables 3-6, indicate that the performance of the algorithms depends significantly on the wind data provided.

Table 3: **Type 1 Constant Wind**

| | Weighted Average | Average | Conservative | Enthusiastic |
|---|---|---|---|---|
| Average Height | 10.67m | 10.59m | 10.70m | 10.57m |
| % Contained | 100% | 100% | 100% | 100% |
| Density Ratio | 0 | 0 | 0 | 0 |

Table 4: **Type 2 Smooth Wind**

| | Weighted Average | Average | Conservative | Enthusiastic |
|---|---|---|---|---|
| Average Height | 11.95m | 12.43m | 12.07m | 20.20m |
| % Contained | 100% | 100% | 100% | 100% |
| Density Ratio | 0 | .90 | 0 | 10321 |

Table 5: **Type 3 Gusty Wind**

| | Weighted Average | Average | Conservative | Enthusiastic |
|---|---|---|---|---|
| Average Height | 11.7m | 12.54m | 11.98m | 19.90m |
| % Contained | 100% | 100% | 100% | 99% |
| Density Ratio | 0 | 0 | 0 | 1357 |

Table 6: **Hurricane Floyd type wind!**

| | Weighted Average | Average | Conservative | Enthusiastic |
|---|---|---|---|---|
| Average Height | 3.34m | 3.48m | 3.33m | 3.32m |
| % Contained | 99% | 98% | 99% | 98% |
| Density Ratio | 11.95 | 42.25 | 34.11 | 49.55 |

Table 7: Comparison of algorithm performance under conditions at several major cities, (Average height, % of water contained, and the ratio of the maximum water density outside the pool to our threshold are the three measurements we consider respectively). ∗ When too much water spills out of the fountain, water densities become too computationally intensive to compute. However, it is clear that when this happens that the fountain is operating well outside of acceptable parameters.

| City | | Weighted | Average | Conservative | Enthusiastic |
|---|---|---|---|---|---|
| | Height | 10.35m | 10.62m | 5.03m | 20.67m |
| **Seattle** | % contained | 99% | 99% | 100% | 75.6% |
| | Desity ratio | 60.7 | 25 | 0 | ∗ |
| | Height | 10.34m | 7.65m | 5m | 20.93m |
| **Chicago** | % contained | 99% | 99% | 99% | 62% |
| | Density ratio | 1357 | 2467 | 21.8 | ∗ |
| | Height | 7.57m | 7.91m | 2.38m | 20.99m |
| **Boston** | % contained | 98% | 97% | 100% | 95% |
| | Density Ratio | 1964 | 11000 | 0 | ∗ |
| | Height | 7.56m | 10.52m | 5.91m | 10.21m |
| **L.A.** | % contained | 99% | 99% | 100% | 91% |
| | Density Ratio | 2196 | 0.396 | 0 | ∗ |
| | Height | 8.66m | 10.21m | 7.74m | 20.84m |
| **D.C.** | % contained | 99% | 99% | 100% | 92% |
| | Density Ratio | 3.36 | 17.7 | 0 | ∗ |

## 7.2 Analysis of Strengths and Weaknesses

There are a couple of caveats to consider before beginning this analysis:

- **The time window of the wind signal in which the simulation is run can affect results.** A gusty day may have a few calm periods. For this reason, all simulations use the same window of the same signal
- **The algorithms should be compared relatively**, rather than absolutely.

All the algorithms performed equally well under constant wind conditions. This is no surprise–the distinguishing features of the algorithms are the way they incorporate knowledge of the wind history. Averages of wind speed, maximum wind speed and current wind speed are all equal when the wind is constant. However, the algorithms do have unique strengths and weaknesses when wind conditions are different.

• The *Enthusiastic Algorithm* consistently achieved the most spectacular fountain heights, but this comes at a cost. Since it only considers the current wind reading, it is always caught by surprise by sudden wind gusts or any increase in wind speed. Except in the constant wind case, the enthusiastic algorithm systematically resulted in too much water being sprayed outside the fountain.

• The *Conservative Algorithm* behaves as conservatively as its name suggests. By always considering the maximum wind speed measured in the recorded wind history, it always had the most paranoid estimate of how bad the wind could get. With this algorithm, all the water is usually contained in the fountain except in rare cases where sudden gusts greatly surpass the maximum recorded wind speed before the next measurement is made. However, the fountain height is often disappointingly low compared to the other algorithms, especially when a large gust of wind was recorded in the wind speed history. Noise in the anemometer signal could erroneously cause a such a spike to be recorded and give this algorithm poor performance.

• The *Weighted Average Algorithm* performs about as well as the *Averaging Algorithm*. Both manage to effectively contain most of the water. However, they are often surprisingly conservative. In some cases such as in the Gusty Wind simulations, the Weighted Average Algorithm was even more conservative than the Conservative Algorithm. The explanation for this is that since both averaging algorithms consider the standard deviation of previous wind speed data, they become more conservative when recent wind speeds are found to be highly variable. But if wind speeds make a fairly sudden change in their general behavior, as in the Hurricane Floyd case, the Weighted Algorithm reacts slightly faster than the simple Averaging Algorithm.

Table 8: Results of tilting the fountain into the wind.

| Wind speed | Maximum height (tilt angle) | |
|---|---|---|
| | without titling | WITH tilting |
| 2 m/s | 16.4 m | 31.0 m (37.5°) |
| 5 m/s | 10.8 m | 22.5 m (8.5°) |
| 7 m/s | 5.9 m | 12.7 m (32.0°) |

# 8 Other Considerations

## 8.1 Fountains with Tiltable Nozzles

Water jets with directional control exist (firefighters use them extensively!). Therefore, given a steady wind, it is conceivable that aiming the fountain slightly into the wind may allow for a higher water stream without additional water spraying outside the fountain pool. We explore this possibility here.

For a range of constant wind speeds, we simulate the fountain at various tilt angles and find the angle which maximizes the fountain height without unacceptable spray landing outside the pool. For each run of the simulation we fire enough blobs so that results are statistically significant (this required ten initial water blobs to be fired from the nozzle). The results are reported in Table 7.

These results show that the fountain can be made nearly twice as high by directing the nozzle into the wind. This would appear very encouraging indeed, were it not for two important points–

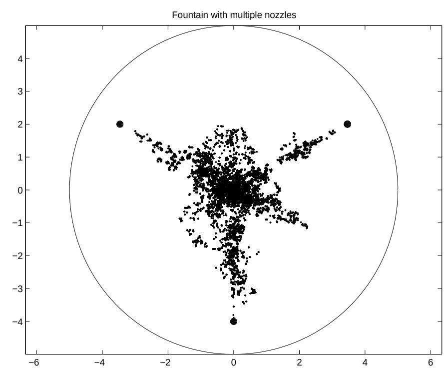
- **The spray distance is extremely sensitive to the tilt angle.** Variations of a single degree in the tilt angle caused unacceptable amounts of water to fall outside the pool area.
- **Real wind is rarely so constant.**

Given the sensitivity of the spray to the tilt and wind, we consider it entirely infeasible to use tilting to increase the fountain height.

## 8.2 A Fountain with Multiple Nozzles

Many ornamental fountains have multiple nozzles. Our model can easily be extended to handle such a situation by superimposing, provided that the stream-stream interaction is not significant. Blasting two streams into one another would require a modification of our fountain model to include a description of this interaction. Most fountains however, do not include any such violent effects and our model is appropriate. Figure 9 shows an example of the spray pattern from a fountain with multiple nozzles: one large nozzle pointing upwards and three smaller nozzles on the edge of the pool pointing inwards. Analysis for the effects of wind can be carried out exactly in the same fashion as the single nozzle case.

Figure 9: The water spray patter for a fountain with one large central jet, and three smaller jets aims towards the center. The large circle represents the outer edge of the pool.




## 8.3 Alternate Pool Geometries

Fountains with non circular pools may also be considered. We can still apply our measure of wetness outside of the pool area by taking the percentage of the water that landed outside of the pool and requiring that no region gets too wet. If the fountain is located in a city where the wind is predominantly in one direction (WebMET data shows that many cities are in this category) then an ellipse shaped pool, with its major axis parallel to the wind direction may work better. In such cases, however, algorithms must weigh the different wind directions accordingly to determine the appropriate safe nozzle velocities.

It is important to note that variation in wind direction can no longer be ignored when considering asymmetrically shaped pools.

## 8.4 Multiple Anemometers

If several anemometers are set on top of nearby buildings, we can gain more insight into the nature of the local area wind conditions and how the wind varies spatially across the region. We may approximate the wind at any point by interpolating these readings across space. We also gain insight to the variation of wind across space. Depending on how consistent the readings are at different locations, we can set the algorithms to be more conservative or less conservative.

## 8.5 Possible Extensions and Further Considerations

There are many factors which contribute to the fountain analysis. We have considered only those effects which we judge to be most crucial.

Other effects that could have been considered include:

- We made many simplifications in order to make the problem tractable. There are parameters that we did not incorporate in our model that may have effect in real life. Some of these include temperature, barometric pressures, etc.
- If there are any indications that a storm is approaching, the fountain should be turned off.
- If the temperature is low, we might set the algorithms to be more conservative, because it is very unpleasant to be wet in cold weather, and ice formation can be dangerous.
- If the buildings around the plaza are significantly closer to the fountain than the 40 meters considered in our simulations, then the dynamics of the wind near the fountain may be altered with the addition of eddies and other turbulence.
- For larger scale fountains that reach heights significantly higher than the nearby buildings, the magnitude of the wind will grow stronger farther above the plaza.
- A longer wind history could be incorporated into the algorithm–perhaps several hours or several days worth of data. This could be used to make a better judgment of potential changes in the wind.

# 9 Reccomendations and Conclusions

Recommendations for which algorithm should be used depends largely on the purpose of the fountain. Tolerances for the acceptable water spray in the plaza may be higher or lower than what we have assumed. If keeping the water spray contained in the fountain pool is a much larger concern than shooting the fountain high into the air, then the Conservative Algorithm may be the best choice for controlling such a fountain. Conversely, if water spray outside the fountain is not an overriding concern, than the Enthusiastic Algorithm may in fact be the best choice.

For a reasonable balance between safety and dazzle however, the Conservative algorithm and the Enthusiastic algorithm are both totally inadequet.

• Either the Weighted Average Algorithm or the Averaging Algorithm should be used.

The Weighted Average Algorithm responds faster to sharp changes in wind speed and therefore will perform better in places like Chicago where wind gusts are more frequent. However, if wind variations are fairly smooth, as in Los Angeles data, then the Averaging Algorithm is the best choice.

## 10 References

## References

[1] Stephen D. Fisher, *Complex Variables* 2nd ed., Dover 1990.

[2] WebMET *The Meteorological Resource Center*, www.mebmet.com, 2001, Accessed Feb 11th, 2002.

[3] Matheos Santamouris and Elena Dascalaki, *Wind Speed in the Urban Environment*, www.brunel.ac.uk/research/solvent/pdf/report3.pdf, Accessed Feb. 10th, 2002.

[4] *Industrial Weather Products Catalog*, www.scientificsales.com, Accessed Feb. 10th, 2002.

[5] www.riverscape.org, Accessed Feb. 10th, 2002.

[6] www.switzerland.isyours.com, Accessed Feb. 10th, 2002.

[7] www.itu.int/brsoft/terrestrial/hf-propagation/GE/gen.htm, Accessed Feb. 10th, 2002.

[8] *National Climatic Data Center*, www.ncdc.noaa.gov, Accessed Feb. 10th, 2002.

[9] Nobauer, Gerhard Thomas *Droplet Motion and Break-up*, University of Oxford Computing Laboratory-Mathematical Institute, 1999

[10] *Spray Nozzles*, www.mrpressurewasher.com/spraynozzles.html, Accessed Feb. 9th

[11] Virtual Science Center Project Team, *ScienceNet* 1997.

[12] NASA, *Re-Living the Wright Way*, wright.wrc.nasa.gov, Accessed Feb 11th, 2002.

[13] Ross, Sheldon M. *Introduction to Probability Models* Seventh Edition, 2000