A Foul-Weather Fountain

Ryan K. Card
Ernie E. Esser
Jeffrey H. Giansiracusa
University of Washington
Seattle, WA

Advisor: James Allen Morrow

Introduction

We 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.

We construct a model of a fountain based on the physics of falling water droplets considered as a particle system. We 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 estimate how anemometer readings from a nearby rooftop relate to 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 general abstract wind conditions. Then we construct a wind signal generator that simulates the conditions of several major cities from meteorological database data, and we compare the performance of our control systems in each city.

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 Los Angeles and Seattle.

The control algorithm cannot possibly respond to changes in conditions at anything below the 10 s scale, since wind is highly variable and the response of the anemometer is somewhat slow [Industrial Weather Products 2002]. The goal is therefore to design the algorithm to operate on a time scale of 10 s up to a couple of hours and adapt the height of the fountain to a maximum safe level.

The UMAP Journal 23 (3) (2002) 251–266.

Model of the Water Jet

We 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 is based on a physically realistic model of small droplets (around 1 mm radius) and we scale it up to an effective model for larger clumps of water (up to 10 cm across) because the physics of turbulence and viscosity at the larger scale cannot be computed accurately.

We need the following assumptions:

- The drag force is proportional to the square of the speed and to the square of the radius [NASA 2002].
- Droplets break into smaller droplets when subjected to wind. Breakup rate is proportional to relative wind speed and surface area [Nobauer 1999].
- When a droplet breaks, turbulence causes the new droplet fragments to move slightly away from their initial trajectory.

Modeling a Single Droplet

We formulate the motion of a water droplet as

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

where $\vec{v}$ is the velocity, $\vec{w}$ is the wind velocity relative to the motion of the droplet (wind vector minus velocity vector), m and r are the droplet's mass and radius, and η is a constant of proportionality. According to the Virtual Science Center Project Team [2002], a raindrop with radius 1 mm falls at a terminal velocity of 7 m/s; so we determine that $\eta = 0.855$ kg/m^3. Large drops fall quickly; very tiny drops fall very slowly, mimicking a fine mist that hangs in the air for a long time.

We assume droplet breakup is a modified Poisson process, with rate

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

If the breakup rate did not depend on variable parameters $|w|$ and r^2, the process would be a standard Poisson process. We determine λ_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 occurs, we split the droplet into two new droplets and divide the mass randomly, using a uniform distribution. Air turbulence tends to impart to the two new droplets a small velocity component perpendicular to the relative wind direction $\vec{w}$. This effect causes a tight stream of water to spread out as it travels, even under zero-wind conditions. We let this velocity nudge have magnitude 2% of the particle's speed relative to the air and a random perpendicular direction. We give the two drops equal and opposite nudges.

Putting Water Drops Together to Make a Fountain

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 (**Figure 1**).

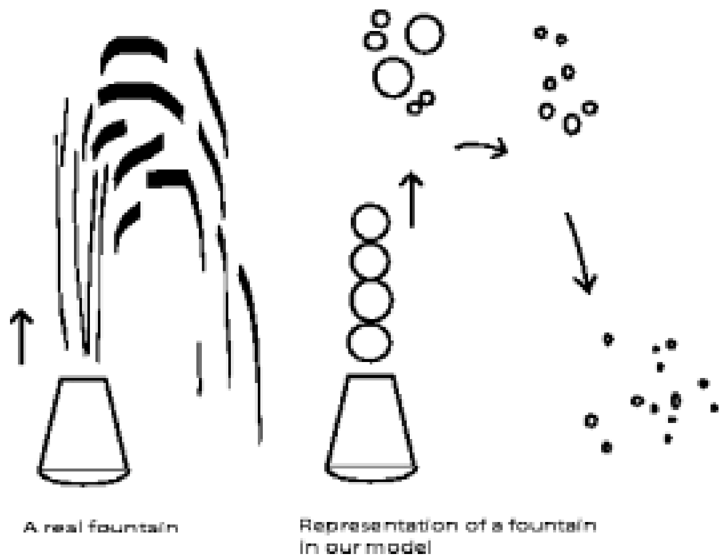

Figure 1. A continuous water jet is approximated by a discrete stream of water blobs.

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 speed:

- The standard deviation of blob radii is 10% of the nozzle size.
- The standard deviation of initial speeds is 5% of the initial speed.
- The blobs leave with an angular spread of 3°, consistent with industrial high-pressure nozzles [Spray Nozzles 2002].

Wind drag in particle streams is significantly reduced for particles following one another closely (NASCAR drivers and racing cyclists are intimately familiar with this phenomenon). These effects are already incorporated into the dynamics of large water blobs (which can be thought of as representing many small drops moving together). We therefore consider this to be an effective model for large drops rather than a realistic interaction model.

Fitting the Fountain

The Five Rivers Fountain of Lights in Daytona, Florida, is one of the largest fountains in the world. 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 in Geneva, Switzerland, another impressive fountain, 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) returns to earth approximately 35 m downwind from the nozzle [Micheloud & Cie 2002] (**Figure 2**).

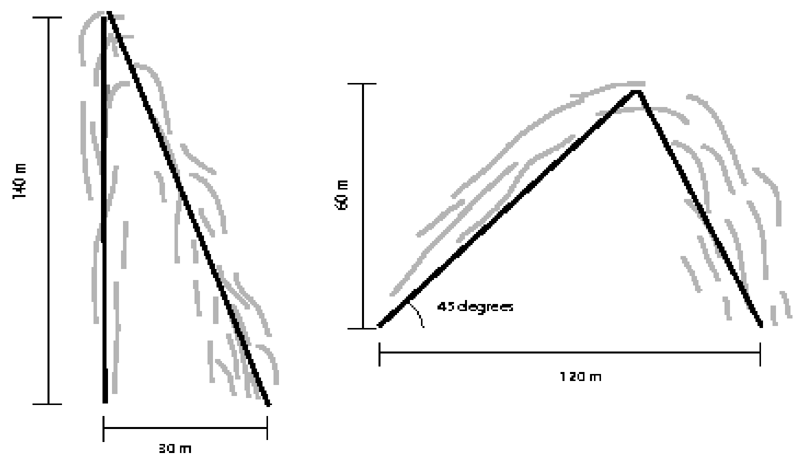

Figure 2. The Jet D'Eau and the Fountain of Lights.

To determine λ_0, we first match our geometry to the Five Rivers Fountain of Lights. We fix λ_0 so that with an initial velocity such that the stream reaches a height of 60 m, it returns to the ground at a distance of just over 100 m. Too large a λ_0 results in the water breaking up too quickly into tiny droplets, which have a much lower terminal velocity and thus fail to reach the desired distance; if the value is too small, then an unrealistically small amount of spray is produced and the water blob travels too far. The results are summarized in **Table 1**.

We set $\lambda_0 = 5000$. The results are highly insensitive to this parameter; varying λ_0 by a factor of 2 cause only a 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.

Table 1.
Comparison between real fountains and our model.

	Jet D'Eau		Five Rivers Fountain	
	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 extreme fountains accurate to within about 15%. We expect that for a plaza-sized fountain, our model will be more accurate, since our formulas for breakup and drag force are derived under less extreme conditions.

Wind Flow Through the Plaza

Buildings and other structures in an urban environment can cause significant disturbances to wind flow patterns; rooftop and street-level conditions can often be quite different, so readings from a rooftop anemometer could be biased. To model the plaza wind, we assume:

- There are no significant structures between the buildings beside of the plaza.
- The plaza is large, so 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.

Formulation

We approximate the geometry of the plaza as in **Figure 3** and use complex analytic flow techniques [Fisher 1990, 225].

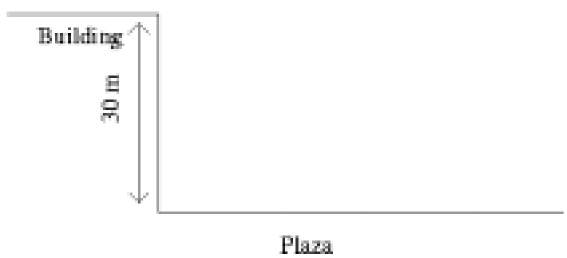

Figure 3. Schematic representation of the relevant features of the plaza.

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:

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

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.

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

$$v_x = \mathrm{Im}\left[\frac{\partial \Gamma_c}{\partial c}\right].$$

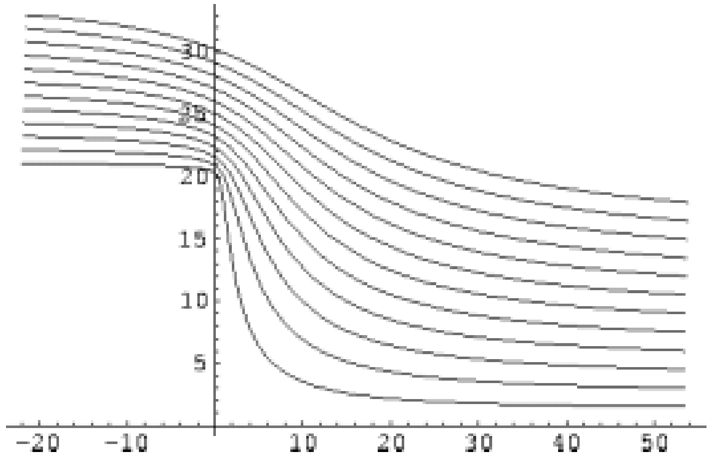

Figure 4. Streamlines for wind flow entering the plaza; decreased wind speed at the plaza level is apparent. Note the highly increased wind speed near the edge of the building.

The horizontal velocity profile for a streamline that passes about 3 m above the building roof (corresponding to $c = 0.6$) is plotted in **Figure 5**; 3 m 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.

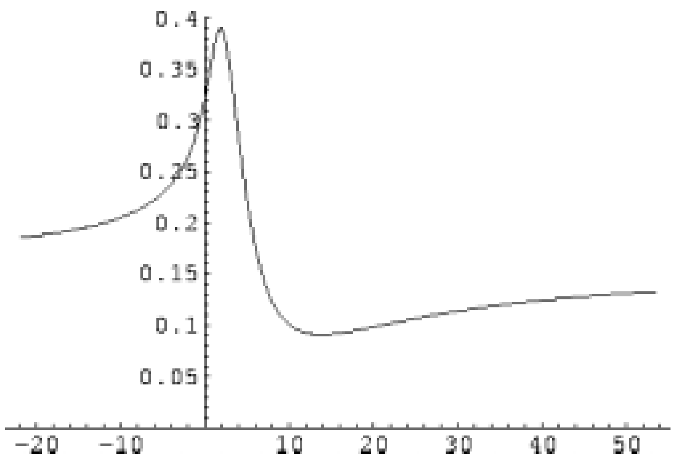

Figure 5. Horizontal velocity profile for the streamline corresponding to $c = 0.6$. This streamline passes above the building's roof at a height of 3 m, a reasonable anemometer mounting height.

This calculation is validated by its excellent agreement with the findings of Santamouris and Dascalaki [2000], who report that in flows perpendicular to a street the ground-level speeds are between zero and 55% of the free-stream speeds.

Results

We conclude 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.
- **The anemometer reports a wind speed that is highly biased!** Plaza-level wind moves approximately half as fast as the roof-level wind.
- **Wind speeds are spatially constant within the plaza airspace.** If the fountain is not significantly higher than the surrounding buildings, then spatial wind variation can be safely ignored.

Modeling Wind Variation Over Time

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

- **Type 1: A low intensity constant breeze of a few m/s,** meant to test the algorithm's ability to judge the proper height for a given wind speed.
- **Type 2: A breeze varying smoothly over a timescale of a couple of 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 tests the algorithm's 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 the gusty scenario can easily fool a naive algorithm.

Generating a Realistic Wind Signal

We parametrize the wind profile of a location by four numbers:

- The **mean steady wind** μ_{steady}.
- The **mean gust strength** μ_{gust}, where a gust is defined to be variation on the sub-15 s timescale.
- The **mean gust duration** t_{gust}.
- The **gust deviation** σ_{gust}.

From WebMET data [2001], we estimate these characteristic numbers for some major U.S. cities (**Table 2**).

We construct realistic wind signals from these characteristic numbers to correspond to our types:

Table 2.
Characteristic parametrization of several major U.S. cities. These parameters specify the plaza wind conditions, which are slightly milder than the free-stream conditions.

	μ_{steady} (m/s)	μ_{gust} (m/s)	t_{gust} (s)	σ_{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 DC	1.3	3.4	3.0	1.0

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

Figure 6 shows a comparison of wind signals for Seattle and Chicago; the extreme gustiness of the "Windy City" is apparent.

We also create a "Hurricane Floyd" wind profile by multiplying a Chicago wind signal by a factor that damps the wind to zero early on (the calm before the storm) and then amplifies it to hurricane level over a period of 10 min.

Fountain Control Algorithms

The goal of the control algorithm is to respond to the anemometer data by maximizing the height of the fountain while minimizing the probability of the plaza area outside the fountain pool being drenched. The control algorithm has access to anemometer readings and direct control over the nozzle speed.

The control system must have some knowledge of how the water spray travel distance relates to nozzle speed and wind speed. We develop two complementary measures of spray distance and tabulate the relationship between them and nozzle/wind conditions. The algorithms that we develop combine this table with an estimate of possible future wind speed (based on the current wind and a stored recent history) to decide on a good nozzle speed.

Measures of Water Spray Distance

Our measures of spray distance are

- the radius within which 99% of sprayed water lands, and
- a threshold for acceptable water density outside the fountain that corresponds roughly to a light rain: 1 cm in 10 h (2.8×10^{-4} mm/s).

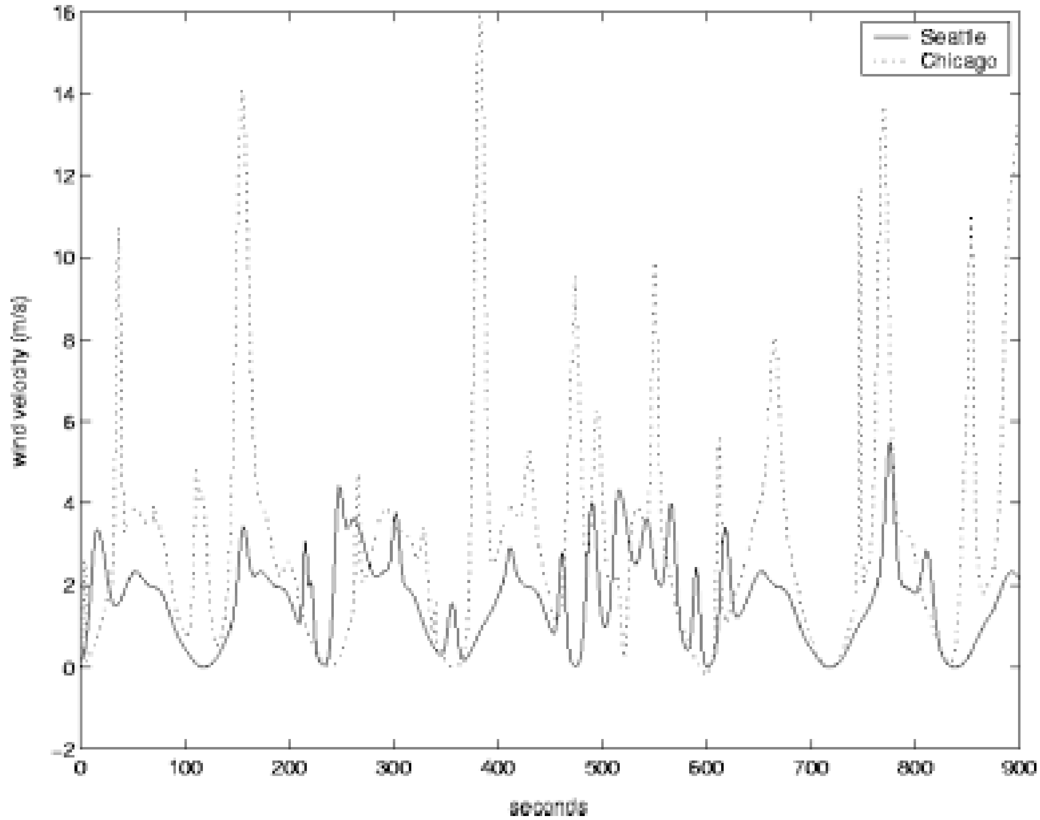

Figure 6. Horizontal velocity profile.

In simulations over a suitably long time period, we find that these two measures agree to within 1%.

We 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 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 could.

Constructing the Control System

We begin with a few useful assumptions:

- Variation in wind direction can be safely ignored. We use the triangle inequality: If the wind pushes a drop first in one direction and then in another, it will necessarily land nearer to the fountain than if it had been pushed in one direction continuously.
- The algorithm has access to real-time anemometer data averaged over 10-s intervals as well as at least a 10-min history of measurements. Even if the anemometer responds faster than 10 s, it is nonsensical to vary the fountain any faster than this, because the water requires approximately this much time to complete its flight.

- For concreteness, we focus on the plaza configuration of **Figure 7**. Most importantly, the fountain is at the center of a circular pool of radius 5 m.

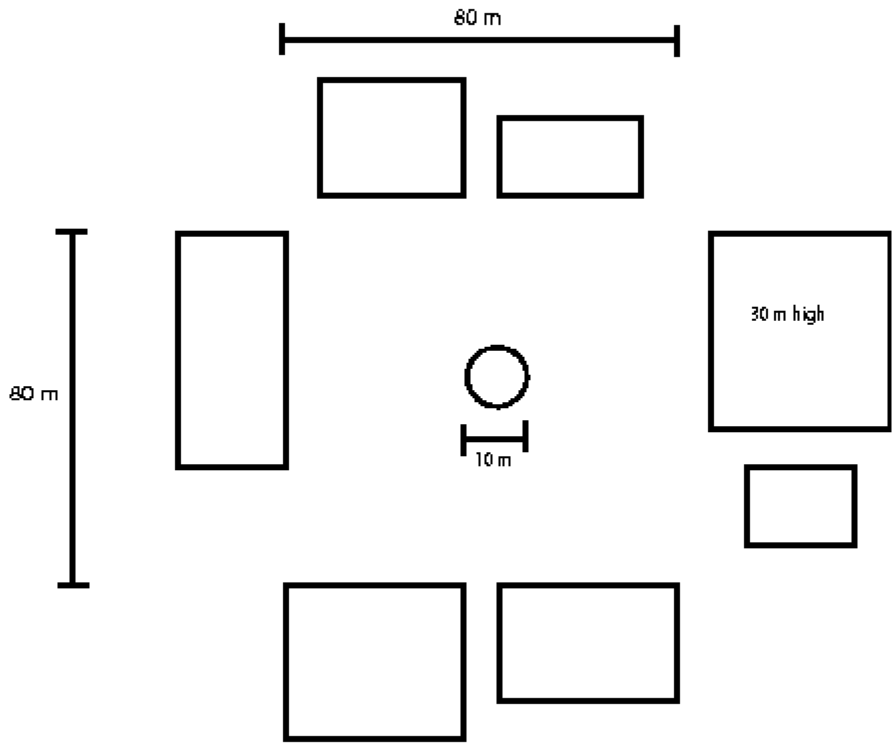

Figure 7. The layout of our hypothetical model plaza.

We display the spray distance as a function of wind speed and nozzle speed in **Figure 8**.

An estimate of how far a water droplet can travel starting at height z_0, falling at its terminal velocity v_t, and moving at horizontal wind speed w is

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

The smallest droplets that our simulations produce have radii of about 1 mm with corresponding terminal velocity 7 m/s. For specific heights and winds, we find that this rough estimate is usually within 30% of the corresponding minimum safe distance shown in **Figure 8**, a good indication that our simulations produce reasonable results.

The Control Algorithms

We formulate four control algorithms:

- **Averaging Algorithm:** This algorithm considers an average of the previous 10 min 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.

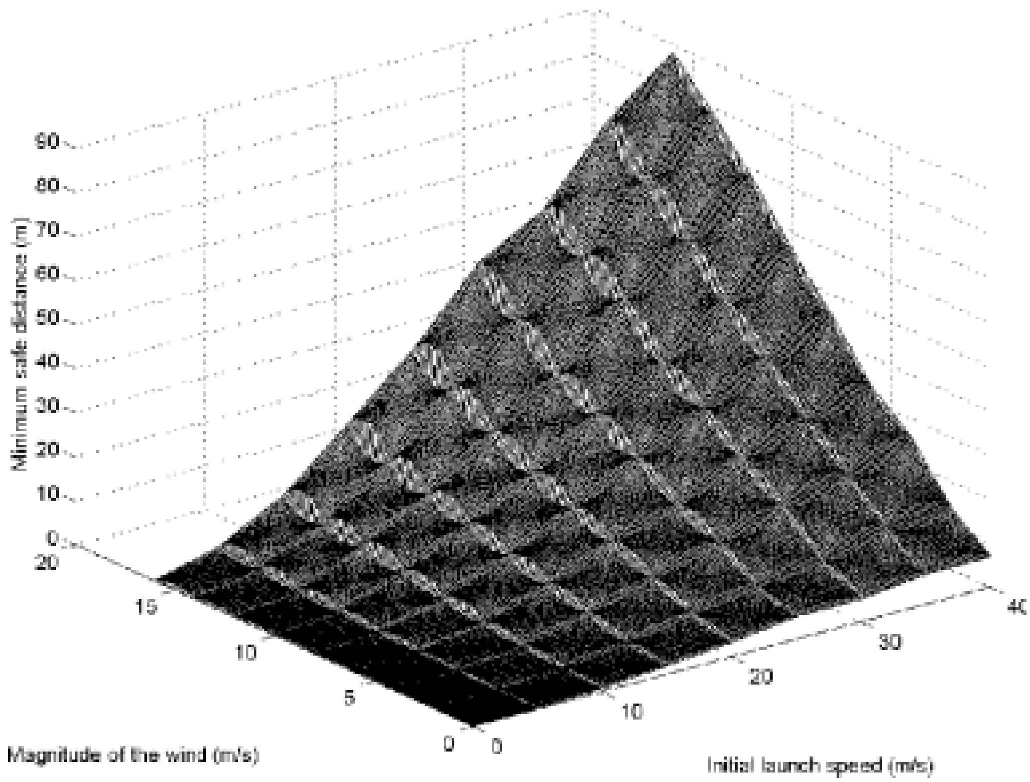

Figure 8. Linearly interpolated table of spray distance as a function of wind speed and nozzle speed. Each data point represents a burst of 5 nozzle-size water blobs.

- **Weighted Average Algorithm:** The key feature of this algorithm is that the data of the last 10 min are weighted linearly according to recentness. The current measurement gets the highest weight.
- **Conservative Algorithm:** This algorithm uses the maximum wind speed measured over the last 10 min to predict the worst-case wind. This is the most conservative approach—it will always err towards safety.
- **Enthusiastic Algorithm:** This algorithm ignores previous wind data history and puts the fountain to the maximum safe height given immediate conditions. No precaution is taken with regard to possible future wind behavior.

Results

Comparing the Algorithms

We test each algorithm 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 DC.
- Hurricane Floyd-type winds!

We run several simulations of the fountain, each for 3 min, under the control of each algorithm—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). We use the following criteria for comparing the performance of the algorithms:

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

The results of our simulations (**Table 3**) indicates that the performance of the algorithms depends significantly on the wind data provided.

Strengths and Weaknesses

All of the algorithms perform equally well under constant wind conditions, but each has unique strengths and weaknesses.

- The *Enthusiastic Algorithm* consistently achieves the most spectacular fountain heights but at a cost. Since it considers only the current wind reading, it is always caught by surprise by sudden gusts or any increase in wind speed. Except in the constant-wind case, the algorithm systematically results in too much water being sprayed outside the fountain.
- The *Conservative Algorithm* always has the most paranoid estimate of how bad the wind could get, and all the water is usually contained in the fountain except in rare cases when 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.
- The *Weighted Average Algorithm* performs about as well as the *Averaging Algorithm*. Both contain most of the water but are often surprisingly conservative. In the Gusty Wind simulations, the Weighted Average Algorithm is even more conservative than the Conservative Algorithm; since both averaging algorithms consider the standard deviation of previous wind speed data, they become more conservative when recent wind speeds are highly variable. But if wind speeds change suddenly, as in the Hurricane Floyd case, the Weighted Algorithm reacts slightly faster than the Averaging Algorithm.

Table 3.

Comparisons of algorithm performance. When too much water spills out of the fountain, water densities become too computationally intensive to compute (denoted by $*$), and the fountain is operating well outside of acceptable parameters.

	Weighted Average	Average	Conservative	Enthusiastic
	Type 1: Constant Wind			
Average height	10.7 m	10.6 m	10.7 m	10.6 m
% contained	100%	100%	100%	100%
Density ratio	0	0	0	0
	Type 2: Smooth Wind			
Average height	12.0 m	12.4 m	12.1 m	20.2 m
% contained	100%	100%	100%	100%
Density ratio	0	.90	0	10321
	Type 3: Gusty Wind			
Average height	11.7 m	12.5 m	12.0 m	19.9 m
% contained	100%	100%	100%	99%
Density ratio	0	0	0	1357
	Hurricane Floyd-type wind!			
Average height	3.3 m	3.5 m	3.3 m	3.3 m
% contained	99%	98%	99%	98%
Density ratio	12	42	34	505
	Seattle			
Average height	10.4 m	10.6 m	5.0 m	20.7 m
% contained	99%	99%	100%	75.6%
Density ratio	61	25	0	$*$
	Chicago			
Average height	10.3 m	7.7 m	5.0 m	20.9 m
% contained	99%	99%	99%	62%
Density ratio	1357	2467	22	$*$
	Boston			
Average height	7.6 m	7.9 m	2.4 m	21.0 m
% contained	98%	97%	100%	95%
Density ratio	1964	11000	0	$*$
	Los Angeles			
Average height	7.6 m	10.5 m	5.9 m	10.2 m
% contained	99%	99%	100%	91%
Density ratio	2196	0.4	0	$*$
	Washington, DC			
Average height	8.7 m	10.2 m	7.7 m	20.8 m
% contained	99%	99%	100%	92%
Density ratio	3.36	18	0	$*$

Possible Extensions

Tiltable Nozzles

Water jets with directional control exist (firefighters use them extensively!). So, with a steady wind, aiming the fountain slightly into the wind may allow for a higher water stream without additional water spraying outside the pool.

For a range of constant wind speeds, we simulate the fountain at various tilt angles and find the angle that maximizes fountain height without unacceptable spray landing outside the pool (**Table 4**). For each run, we fire enough blobs (10) so that results are statistically significant.

Table 4.
Results of tilting the fountain into the wind.

Wind speed (m/s)	Maximum height (m) no tilt	tilt	angle
2	16.4	31.0	37.5°
5	10.8	22.5	8.5°
7	5.9	12.7	32.0°

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 cause unacceptable amounts of water outside the pool area.
- **Real wind is rarely so constant.**

We therefore consider it infeasible to use tilting to increase the fountain height.

Multiple Nozzles

Our model can be extended to handle multiple nozzles by superimposing, provided that the stream-stream interaction is not significant.

Alternative Pool Geometries

We can handle fountains with noncircular pools, measuring the percentage of water that lands outside of the pool and requiring that no region gets too wet. If the fountain is in a city with wind predominantly in one direction, then an elliptical pool with major axis parallel to the wind direction may work better, though variation in wind direction can no longer be ignored by the model.

Other Considerations

- There are parameters that we did not incorporate in our model that may have effect in real life, such as temperature and barometric pressures.
- If a storm is approaching, the fountain should be turned off.
- At low temperature, 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 m 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 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.

Recommendations and Conclusions

If keeping the water spray contained in the pool is a much larger concern than shooting the fountain high into the air, then the Conservative Algorithm may be the best choice. Conversely, if water spray outside the fountain is not an overriding concern, than the Enthusiastic Algorithm may be best.

For a reasonable balance between safety and dazzle, the Conservative Algorithm and the Enthusiastic Algorithm are both totally inadequate:

Use either the Weighted Average Algorithm or the Averaging Algorithm.

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

References

Fisher, Stephen D. 1990. *Complex Variables*. 2nd ed. New York: Dover.

WebMET: The Meteorological Resource Center. 2001. `www.webmet.com` . Accessed 11 February 2002.

Santamouris, Matheos, and Elena Dascalaki. 2000. Wind speed in the urban environment. `www.brunel.ac.uk/research/solvent/pdf/report3.pdf` . Accessed 10 February 2002.

Industrial Weather Products Catalog. 2002. `www.scientificsales.com`. Accessed 10 February 2002.

Micheloud & Cie. 2002. `www.switzerland.isyours.com`. Accessed 10 February 2002.

NASA. 2002. Re-Living the Wright Way. `wright.wrc.nasa.gov`. Accessed 11 February 2002.

National Climatic Data Center. 2002. `www.ncdc.noaa.gov`. Accessed 10 February 2002.

Nobauer, Gerhard Thomas. 1999. *Droplet Motion and Break-up*. Oxford, England: University of Oxford Computing Laboratory-Mathematical Institute.

Spray Nozzles. 2002. `www.mrpressurewasher.com/spraynozzles.html`. Accessed 9 February 2002.

Virtual Science Center Project Team. 1997. ScienceNet. Accessed 9 February 2002.

Ross, Sheldon M. 2000. *Introduction to Probability Models*. 7th ed. New York: Academic.