Urban Installation of Smog Reducing Materials

New Mexico Supercomputing Challenge

Final Report

April 5th, 2017

Team Number: SFHS3

Santa Fe High School

Team Members:

Rowan Cahill

Lisel Faust

Theo Goujon

Ramona Park

Teacher Sponsor:

Brian Smith

Teacher Sponsor and Mentor:

Hope Cahill

Table of Contents **Page**

Executive Summary

Air pollution is a major contributor to many detrimental health and environmental issues around the world. As such, companies worldwide have been creating products aimed at passively reducing smog concentrations in urban environments, including titanium dioxide bearing construction materials. A continuation of our simulation from last year, this project explores the most effective placement of these materials at various elevations, ranging from sidewalks to rooftops via NetLogo 3D. In an effort to update and improve the model, additional research has been conducted. This has resulted in the integration of several new subroutines, including fluctuations in the amount of smog produced, urban wind patterns, solar progression, and expansion of the cityscape. These additions to the code more accurately portray how effective the materials are at neutralizing smog in an urban setting when placed at distinct heights. The data acquired from our simulation indicates that anything other than full coverage significantly decreases the amount of smog processed. As the expense of these materials can be prohibitive, however, our results suggest that their application at higher elevations in the section of Albuquerque modeled would be best for smog reduction, as well as more cost effective. A combined bottom and middle application is also worth considering, as it is likely to result in a lower concentration of smog at ground level and therefore a healthier environment for residents.

1. Introduction (adapted from previous report)

1.1. Purpose

Over the past few years, smog has become a worldwide problem. In many areas, it disturbs the population's daily lives and activities, ranging in severity from obstruction of long distance vision to fatal illness. As a continuously growing threat, companies globally are inventing new methods and materials to help reduce smog in the air, including titanium dioxide coated products. TiO_2 bearing concretes, paints, and roofing tiles have proved effective in diminishing smog almost instantly; however, at up to six times the expense of ordinary cement they are costly, meaning they are only economically feasible when applied where most impactful. In a continuation of our simulation from last year, our NetLogo 3D model explored the most effective placement of these materials at various building elevations with the intention of determining the most economically and environmentally conducive installation. While our previous iteration enabled us to determine the placement that maximizes smog reduction at a basic level, it did not account for several environmental factors and was rather rudimentary in the design of the cityscape. To this end, the model has been improved to include a more accurate cityscape, fluctuating levels of smog production, and environmental factors such as wind and sunlight that can impact smog and the performance of smog reducing materials.

1.2. Significance

Smog is directly responsible for various health issues in both humans and animals. These effects include, but are not limited to: eye irritation, respiratory issues (including worsening of asthma), cancer, and even death. The World Health Organization (WHO) has estimated that 2.4

million people worldwide die annually from causes directly attributed to smog [1]. In fact, the most affected countries, such as China, experience more than 4,000 casualties a day [2]. Not only does smog negatively affect humans, but it also injures the living world in and around heavily urbanized areas. Larger amounts of smog can inhibit the growth of plants and directly result in the deaths of hundreds of animals [3]. Moreover, because smog is a mixture of various greenhouse gases, it is a known contributor to global warming.

Figure 1: Harbin, China's Heilongjiang province
Source: AFP/Getty Images

On a local level, Albuquerque may not be as affected by smog as cities throughout China or even coastal California, but the city still requires emissions testing and regular air quality monitoring. A recent summary of the 2016 *State of the Air Report* found unhealthy levels of ozone, a component of smog, in the metro area [4]. Additionally, Public Health New Mexico stated that businesses within Bernalillo county area reportedly released over 130,000 pounds of chemicals into the air over the last twenty years, whereas Santa Fe County only emitted 250 pounds during the same period [5]. Local topography and meteorological conditions, including radiational cooling which results from an increase in temperature with height, or nocturnal inversion, exacerbate the problem by trapping air pollution. This is fortified by a "mountain valley wind effect [6]."

1.3. Background

Smog is created when sunlight reacts with nitrogen oxides (NO_xs) and at least one volatile organic compound (VOC) in the atmosphere. VOCs are organic chemicals that exist in a gaseous state at room temperature due to a low boiling point. Examples of VOCs comprise the vapors produced by gasoline at service stations, oil-based surface coatings, solvents, fuel combustion, and some types of vegetation. Similarly, sources of NO_xs include manufacturing industries, electricity generating stations, fossil fuel powered plants, oil refineries, pulp and paper plants, and incinerators [7; 8]. Other compounds present in a smog sample include sulfur dioxide (SiO_2), ozone (O_3), and particulate matter. As you can see, most of the particles that combine to create smog are byproducts of industrial or combustive processes. Therefore, urban areas with significant transportation and industrialization are more susceptible to smog and the problems it creates.

Figure 2: Harbin, China's Heilongjiang province
Source: ChinaFotoPress/Getty Images

Common rush hours, marked by an increased production of these particulates, further compound these issues. These hours typically fall between 6-10 A.M. and 4-8 P.M. in larger cities. A study published in *Atmospheric Environment* concludes, "Considering the combined effect of driving behavior, vehicle volume and mix, emission factors, on- and near-road concentrations of CO, HC, and NO_x are expected to nearly double during rush hour periods as compared to free-flow periods, given similar dispersion conditions [9]." Due, in part, to an uptick in acceleration/deceleration events specific to intersections, this increase is even more significant on arterial roads that feed into freeways and cut through the cityscape.

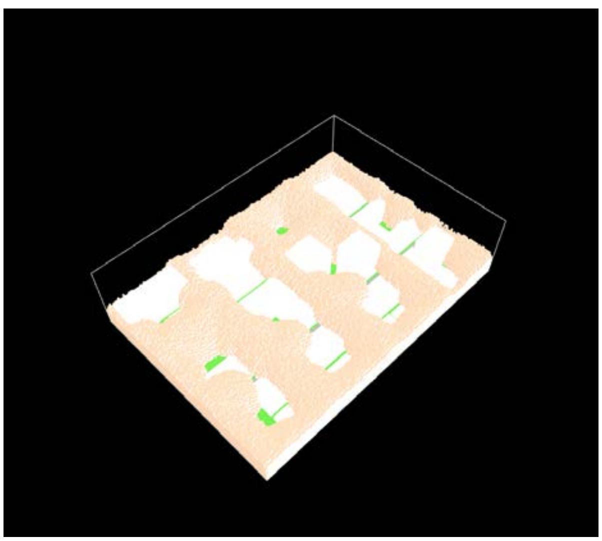
To lessen or entirely neutralize the effects of smog, construction materials, such as concretes, paints, and roofing tiles, are being made with nanoparticles of titanium oxide (TiO_2). When exposed to sunlight, or more specifically the ultraviolet (UV) and visible light within its spectrum, these particles act as a photocatalyst, accelerating the breakdown of the chemicals that make up smog by exciting the electrons within the compound and causing it to be more reactive. The titanium oxide then reacts with adsorbed surface water to create hydroxyl and superoxides. These substances break down the NO_xs that make up smog into relatively harmless byproducts, such as oxygen, water, carbon dioxide, nitrate, sulfate, and calcium nitrate. Ideally these substances are washed off the side of the building or structure with the next rain storm or road washing [11]. If the reaction is incomplete, however, potentially hazardous acids, such as nitric acid may be produced. This is mitigated by calcium carbonate, a base that is present in concrete and asphalt, which helps to neutralize any acidic components that result from a partial reaction. A study conducted by the Belgian Road Research Centre (BRRC) following on site applications of titanium oxide cement shows the product works most efficiently at 5,000 Lux. A nearly 80% reduction in NO_x concentration (ppm) is recorded at this level. At 1,000 Lux, the removal rate drops by half to roughly 40% [12].

A key environmental factor of a city is the winds its gridiron structure creates. One of the most influential of these city currents is the incredibly strong downdraft of wind caused by buildings that are above the average level of rooftops. These much taller buildings catch high velocity winds, which are then sped down the building (much like a sail on a boat) to hit street level. Once there, they are splashed around and aid in the formation of both the M*onroe* and *Vortices Effects* [13]. The *Monroe Effect* is a strong updraft along the sidewalk while the *Vortices Effect* is defined by miniature, street level tornadoes that are caused by buildings with sharp turns or corners. Fast moving wind cannot properly navigate these and cause an eddy of wind near the corner. Additionally, the proximity of buildings within a cityscape can generate winds more than 50 mph, influencing the direction and distribution of smog. This phenomenon, commonly known as “wind tunneling,” is a variant of the *Venturi Effect*. The *Venturi Effect* states that if a large concentration of a substance is forced into a small (or narrow) opening, the pressure exerted upon its exit will be greater; narrow city streets, therefore, result in higher velocity winds [14].

2. Description

2.1. Scope

Our model replicates the dispersal of smog throughout a cityscape and its removal by titanium oxide at varying elevations from ground level to rooftops. Based on our research, the smog originates along the roadway, disperses outward, and fluctuates periodically throughout the day. To indicate the presence of titanium oxide in our model, a white patch coloration has been used. These patches communicate with the surrounding smog patches, representing the interactions between the TiO_2 particles in the concrete and NO_xs in the air. The positioning of the TiO_2 relative to the sun throughout the programed day and shadowing caused by adjacent buildings factors into the amount of smog processed. Via multiple subroutines, the code simulates the smog's diffusion both vertically and horizontally, and models the effect that urban wind patterns have on this diffusion and subsequent removal. The use of patches as opposed to agents allows for a more accurate approximation of a city and modeling of the interaction between the smog and the buildings. To determine the effectiveness of the TiO_2 coating at various elevations, the location of the patches is changeable.



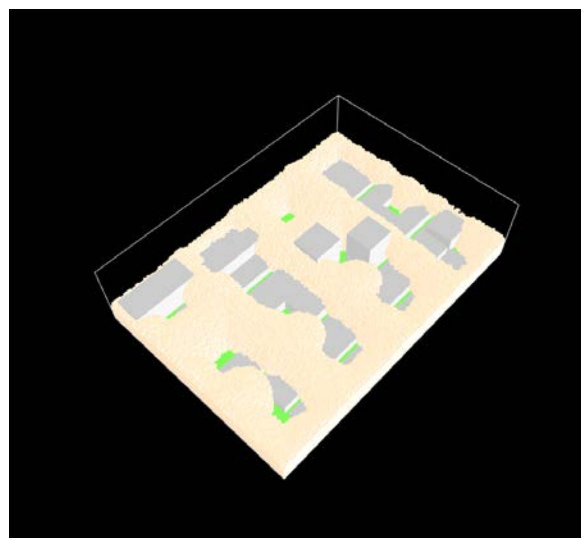


Figure 3: 25 Ticks No Coverage – 2017 Simulation
Figure 4: 25 Ticks Full Coverage – 2017 Simulation

2.2. Scaling and Set-up

In keeping with our previous model, the scaling of the patches and ticks reflects the following research and calculations. Per the Lawrence Berkeley National Laboratory in 2008, one square meter of TiO_2 coated material can remove 200 cubic meters in one 12-hour day [15] under optimal conditions. Therefore, in one hour one square meter removes 16.6 cubic meters. For the rest of the calculation, the scaling of the patches must be taken into consideration. In this simulation, each patch is the equivalent of one story or three meters tall. Due to this, each patch is a 3x3x3 cube and each face has an area of 9 square meters. As such, the multiplication of 16.6 by 9 results in 149.4 cubic meters. When divided by 27 (each cubic patch contains 27 cubic meters) it can be determined that one patch can process 5.53 cubic patches of smog. Through this we can also extrapolate that in 1 / 5.53 (≈11 minutes) of an hour one cubic patch is processed. This data gives us two important pieces of information: one tick represents 1 / 5.53 of an hour and during that time a single patch of smog is processed per TiO_2 face. Additionally, we find that one 12-hour day is roughly 66 ticks or 5.53 multiplied by 12. If left unchecked by a set number of ticks, the dispersal would eventually create a state of equilibrium in which the smog added is matched by the amount lost [16].

In an effort to increase the applicability of the code, this year's simulation was extended to include 12 city blocks and 21 buildings modeled after a section of downtown Albuquerque. This is a significant change from the rudimentary city block and 8 buildings of our previous iteration. The Albuquerque neighborhoods modeled were chosen due to the varying heights of the buildings. The dimensions used were calculated with the following statistics. An average city block has a length and width of 105 meters, which is roughly equivalent to 35 patches. In this model, each block has a width of 34 patches and a length of 36 patches. Each building in the model is the same height and relative shape as the actual buildings in downtown Albuquerque. As previously established, 1 patch is the equivalent of one story. The tallest building in this model is 22 patches and the shortest is 2 patches. The roads are each 4 patches wide.

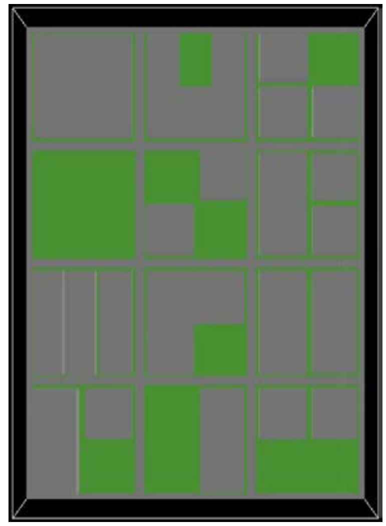



Figure 5: Section of downtown Albuquerque used
Figure 6: Model cityscape

2.3. Methods

The diffusion of smog is one of the most critical subroutines carried over from the first iteration of our model. As the pre-built *Diffusion* script does not function in NetLogo 3D, the *Neighbors6* script, which allows for patches to communicate with their neighbors, is our foundation. At each tick, all patches, excluding surfaces, will ask their neighbors in all directions, if they are air patches and tally their response. It then divides its current concentration (or percentage) of smog by the number of surrounding air patches plus one. When this is done, it resets its smog concentration to this value and available adjacent patches add this number to theirs. As this process continues, the smog disperses throughout the area from its point of origin along the roadway. Based on the tick, smog production increases or decreases to better reflect potential traffic. Rush hour reflects a 100% increase from free-flow time periods.

Urban wind patterns, including wind tunneling, downdrafts, and vortices, newly added to the code this year, influence this diffusion. To simulate these winds, it is necessary to first establish wind direction. This is done with a variable set prior to running the program that reflects one of the 4 cardinal directions—north, east, south, and west. From there, each air patch that is not obstructed by a building, sets their wind variable to a number correlating to that direction. The *Neighbors* script then allows air patches to determine their position relative to surfaces and thus the effect the wind will have on any smog contained. Wind tunnels are created by determining the position of an air patch in relation to the buildings on either side. If it is within 3 patches, half the average distance between buildings, any smog present will move rapidly in the pre-established direction prior to diffusing. For instance, if an air patch has a variable correlated to north and is within 3 patches of a building to the east or west of it, the smog will move north before diffusing. Downdrafts are programmed in a similar fashion. If an air patch is within 1 patch of a vertical surface in the direction of the wind, any smog present will move downwards. These movements prohibit the ability of the smog to diffuse against the wind. Vortices, on the other hand, redirect without restricting. If an air path is diagonal to the corner of a building, the neighboring patch will divert the smog around the corner.

Sunlight, which directly affects the rate at which TiO_2 can process smog, has also been introduced to the current iteration of the code. To determine which sections of a building receive direct sunlight, agents originating from the east, west, and top edges of the model move downward at angles related to the time of day. These angles are calculated at each tick by taking the number of ticks passed and multiplying it by 180/66. This number represents 180 degrees of movement over the 66-tick day. Whenever one of these agents strikes a surface, the patch struck changes a variable to indicate it receives direct sunlight, increasing the rate at which the smog is removed from neighboring air patches by 100%.

2.4. Challenges and Limitations

Throughout the process of creating our simulation we circumvented several challenges. The most significant perhaps was the discussion surrounding the distribution of TiO_2 within the 3 elevations to be tested. Many scenarios were taken into consideration. As Albuquerque is a smaller city with not as many tall buildings, determining the placement based off the tallest building created bias towards the bottom placement. To prevent this, the average height of the buildings was used instead of the maximum height to determine the elevations the TiO_2 was placed at. Then the surface area of each placement was calculated to ensure that roughly the same amount of TiO_2 was applied. However, in the process of ensuring that the bottom was not biased we created a situation in which the top, though almost equal in surface area, has a slight advantage as it occurs over a greater number of stories. The smog, therefore, is exposed for a longer period to the TiO_2. This does not appear to have had a substantial effect on the percentage of smog processed.

A limitation encountered is the collection and representation of the data. As the programming and computational power required to record the concentration of every patch or their average would be too great, the data is represented as a count of cubic meters affected by smog. In other words, the code counts every concentration as equal and tallies each patch containing smog. Similar cubic meter readings of smog in the various trials, therefore, could contain very different concentrations. However, the relative concentration, smog patches to air patches, still accurately represents the spread and dispersal of the smog particles through the city.

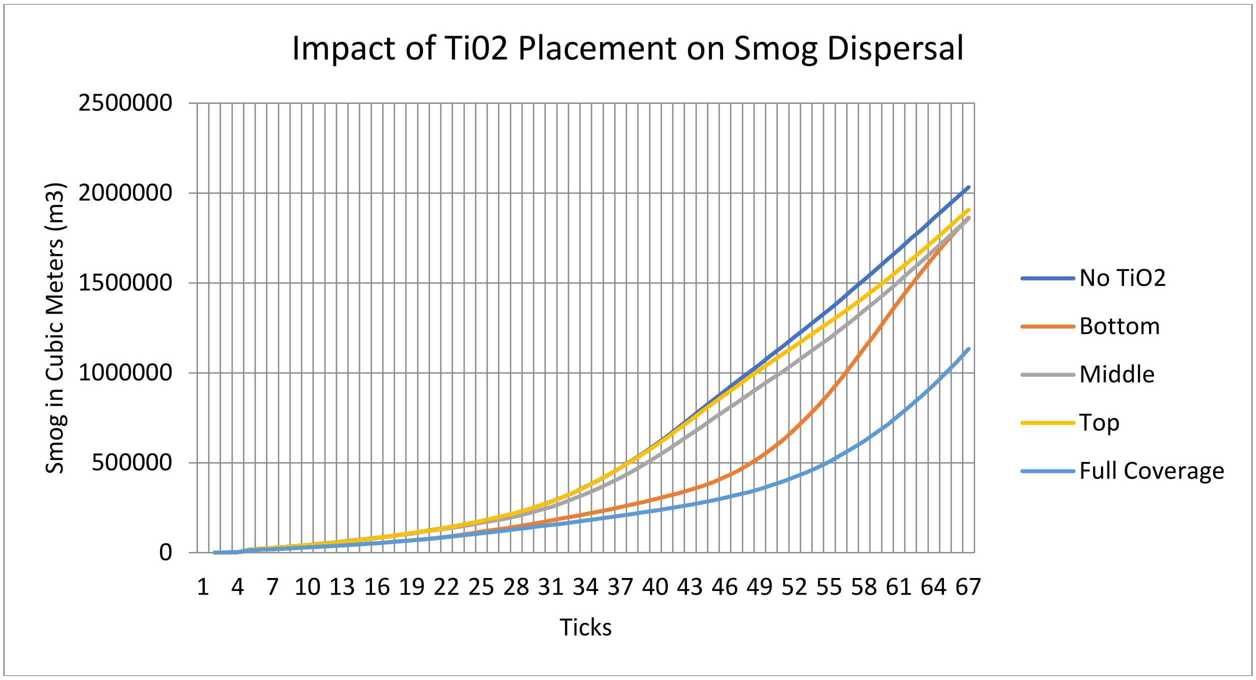
Though our code considers most of the effects on the production, diffusion, and removal of smog, it does not account for every possible environmental variable. For instance, a key factor in titanium oxide's efficiency, the humidity of the surrounding air which causes smog particulates to adhere to the photocatalytic surface [12] is not included. Nor are wind patterns caused by the surrounding landscapes, such as Albuquerque's nocturnal inversions. Given the resource intensive code, requiring a high-speed processor and large amount RAM, any additions are limited; however, by accounting for the other key variables, minimal discrepancy would be expected.

3. Conclusions

3.1. Summary of Previous Conclusions

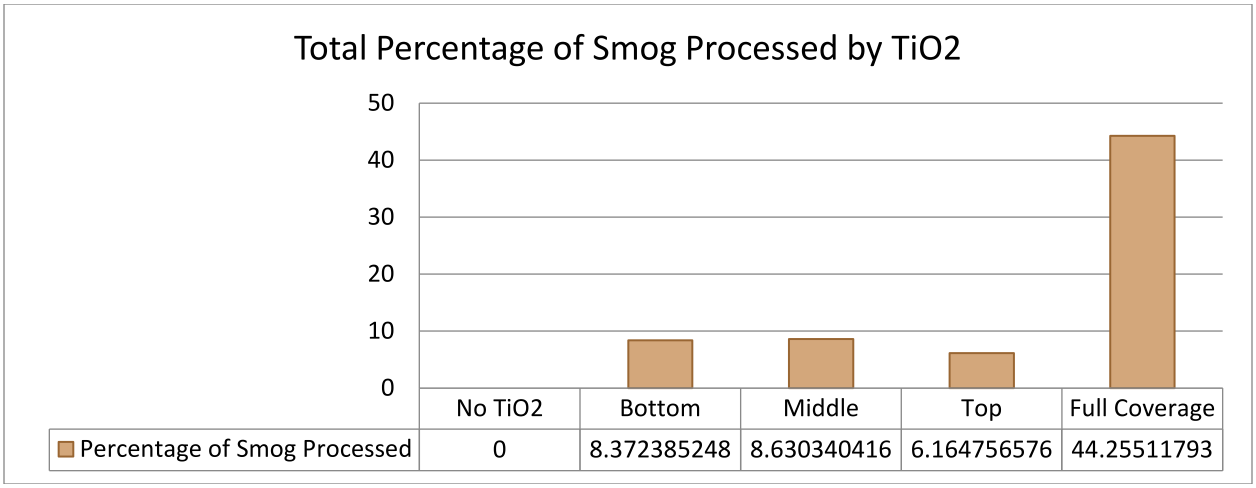
As a continuing project, it is crucial to understand the previous year's simulation and data, as it provided the base for this year's expansion. Our first NetLogo 3D iteration modeled the uniform diffusion of smog along a single city block and its removal by strategically placed white titanium oxide patches. Patches were placed at three building elevations—low, medium, and high—to determine their most effective placement. The data, measured in cubic meters, from these trials was then compared to full and no coverage of the buildings.

A comparison of the percentage of smog processed during each placement of TiO_2 shows a nearly identical removal rate for the bottom and middle placement at 8.4 and 8.6% respectively. The top placement fell short at 6.2%, as the smog had a greater opportunity to build unchecked. These rates pale in comparison to the 44% processed (or removed) by full coverage of the buildings.

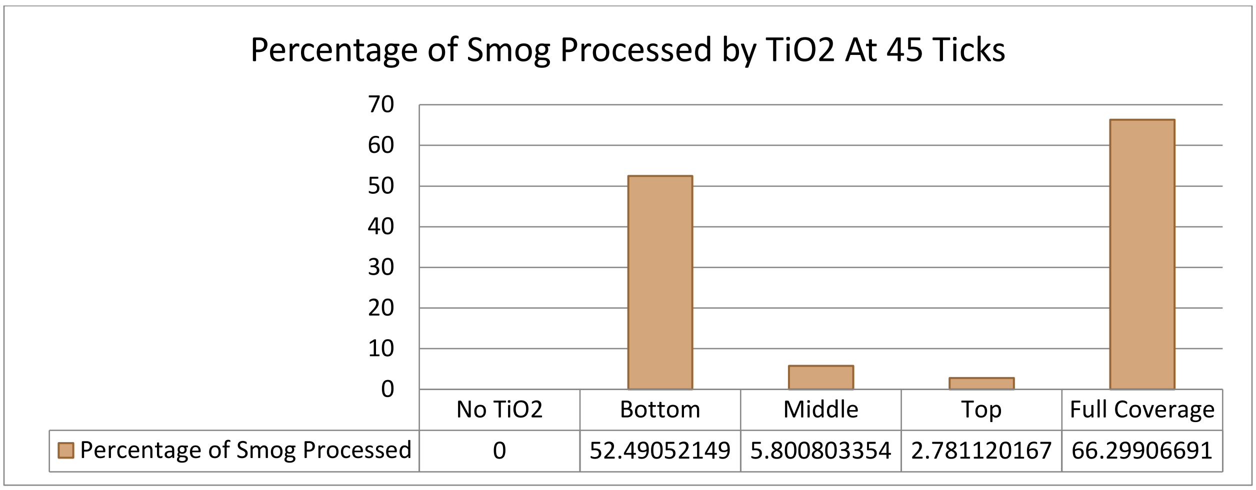


Graph 1: Impact of TiO_2 Placement on Smog Concentration - 2016 Simulation [16]

Further analysis of the bottom and full coverage data sets show similar progressions until roughly the 45th tick. At this point, there was a sharp increase in the difference. This trend reflected the smog surpassing the height of the TiO_2 at which point it diffused unhindered. Clearly full coverage application of titanium oxide bearing materials is most effective for the reduction of smog; however, it is not the most cost effective. A bottom application would be more economically feasible and the second most effective up to a point, roughly mid-day, at which time it was overcome by the smog and falls behind the middle coverage by 2%.



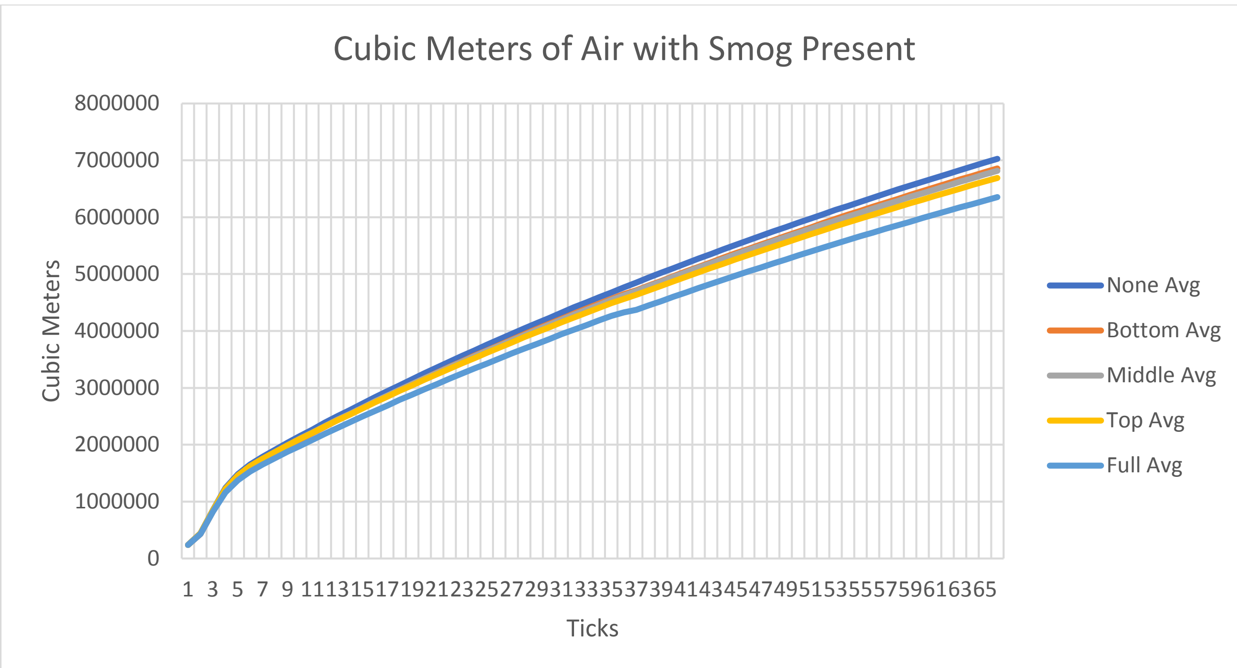
Graph 2: Total Percentage of Smog Processed Per Elevation - 2016 Simulation [16]



Graph 3: Percentage of Smog Processed at 45 Ticks Per Elevation - 2016 Simulation [16]

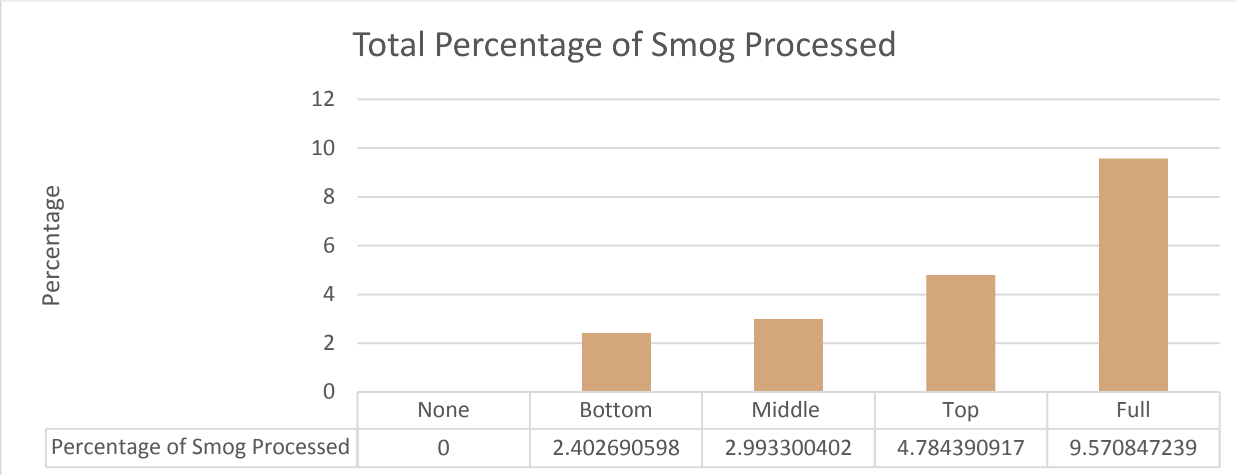
3.2. Results

The overall progression of the smog concentrations recorded is reminiscent of a logarithmic curve, with initial rapid growth followed by slow gains. All trials followed the same trend until roughly the 7th tick, when they separated into three distinct groupings with none and full coverage flanking the bottom, middle, and top placements of TiO_2 respectively. At the 39th tick, a slight deviation between the path of the bottom and middle placements from the top is observed, resulting in a $1.70*10^5$ cubic meter difference between the bottom and middle trials when compared to the top trials. As seen below, after an entire simulated day the trial with no TiO_2 accumulated roughly $7.03*10^6$ cubic meters of smog, followed by $6.86*10^6$ for the bottom, $6.82*10^6$ for the middle, $6.69*10^6$ for the top, and $6.35*10^6$ for full coverage of the buildings.



Graph 4: Cubic Meters of Air with Smog Present - 2017 Simulation

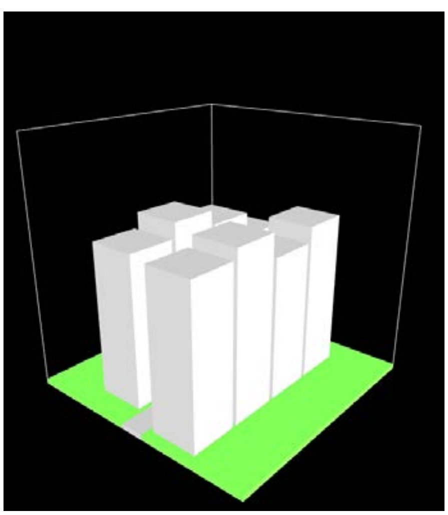
As observed in both graphs, there is a positive correlation between the height of the TiO_2 placement and the amount of smog processed. The total percentage of smog reduced by the top placement proves it to be the most effective of the experimental trials, as illustrated by the graph below. This placement processed exactly half the amount of smog reduced by full coverage of the buildings. The bottom placement divides this in half again. The trials where no TiO_2 was added to the simulation provided the baseline where by which these percentages were calculated.



Graph 5: Total Percentage of Smog Processed - 2017 Simulation

3.3. Analysis

This year's results differ significantly from our previous simulation. There are several changes to the code that could account for this, including the expansion of the cityscape. The previous iteration did not have a large variety of building heights or plazas as found within the section of Albuquerque modeled. Smog diffuses unchecked in these openings, as it is not within the proximity of the titanium oxide.



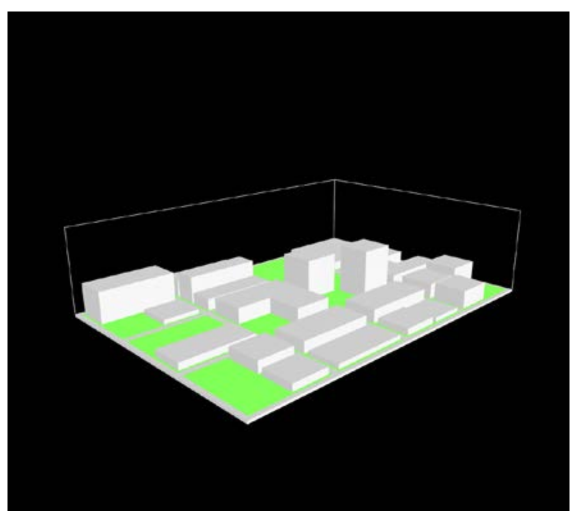


Figure 7: Cityscape – 2016 Simulation [16]
Figure 8: Cityscape – 2017 Simulation

It is likely that the addition of sunlight also affected the results. Creating a city with varying elevations and narrow roads means that at any given point in the simulated day most of the TiO_2 is operating at 50%. Therefore, the percentage of smog processed is significantly less than in the previous model. To compound this, smog is not only spawned at a greater frequency along the roadways in the current program, but there are also more roadways. This means that a larger amount of smog is produced per tick.

Despite these differences, full coverage is still the most effective, removing 9.57% of smog by the end of the day. The top placement deviates from this by 4.78%, indicating that it is the most effective experimental elevation for removing smog in this iteration. While a single patch face can remove 1,800 m^3 (66 patches) of smog over the 66 ticks run, the TiO_2 patches only process smog that is in direct contact with them. Therefore, the patches at higher elevations are ineffective until the smog reaches that height. This allows the smog to build up at lower elevations and remain at higher concentrations. Although the bottom and middle applications remove a lesser amount of smog, 2.40% and 2.99% respectively, it can be safely assumed, that their concentration of smog near ground level would be more tolerable in a real-world scenario. As such, the best placement, exclusive of full coverage, comes down to what the goal of the city planner is: to remove the most smog from the air or keep the lower, commuter levels at healthier levels of smog.

3.4. Future Plans

Though many of the complexities of smog diffusion and its removal by TiO_2 are captured in the simulation, there remain several worthwhile additions. First and foremost, would be the inclusion of humidity to the code, as it, like light intensity, impacts the processing rate. Further expansion of the cityscape and modeling of those cities most impacted by smog are also under consideration. To do this, we would like to develop a code capable of processing 3-Dimensional representations of cities and converting them into patch maps that NetLogo 3D can create. This would allow our model to be customized to any city or condition and dramatically increase its applicability. Lastly, we would like to increase the scope of our data to include actual smog concentrations, allowing us to better analyze the impact of each TiO_2 placement.

Acknowledgements

Thank you to all of those below…

Ms. Hope Cahill, a teacher at El Dorado Community School, provided regular brainstorming, organizational, and editing support throughout the entire project. She made sure that the team kept focus during our meetings and helped us keep on track. This project would not be completed without her.

Mr. Brian Smith was extremely helpful in the initial organization of the team and keeping it on track to finish. He made sure that we had the funds to register for the Supercomputing Challenge and participate in the Kickoff.

Mr. Derek Buschman, our chemistry teacher at Santa Fe High School, was of great assistance in the interpretation of the chemistry involved in the reports used. He made sure we could understand the science behind what we were modeling.

Mr. Simon Boses, the founder of Make Time at La Tienda, El Dorado, who allowed us to use his computer lab to run the code and complete data collection. Without his assistance, it would have been difficult to test the completed code and wrap up the project on time.

Ms. Karen ___ who was incredibly helpful in troubleshooting our code as she provided us with the methods to reset the overhead RAM limit in Netlogo. Her assistance enabled us to gather our data far more expediently than it would have been.

Team Biographies

Rowan Cahill is currently a junior attending Santa Fe High School and has an interest in all things mathematical. The past two years, he and his team were one of nine finalists and received additional accolades for their projects. This year, Rowan created the code controlling the diffusion of the smog and its interaction with the buildings, as well as the subroutines tackling the interactions of wind and sunlight. He also contributed to the research, writing, and final edit of the paper. email: rcahill505@gmail.com

Lisel Faust currently attends Santa Fe High School as a junior and loves to challenge herself by taking the hardest course load possible. She is very passionate about science and technology, and hopes to pursue a career in either of those fields. She discovered her love for coding, during the Santa Fe Institute Complexity Camp in the summer of 2015. For this project, Lisel helped write the code, research the issues, and used her organizational skills to make sure the project advanced on time. email: faust.lisel@gmail.com

Theodore Goujon is currently a junior at Santa Fe High School who is both academically driven as well as a dedicated athlete. He is currently interested in the dynamics of international relations between countries and hopes to join the Foreign Service or some other related field that deals with international cooperation. He has been on the team for three years, and was a part of last year's finalist team. For this year's project, he contributed to the writing of the interim and final reports. email: theogoujon@gmail.com

Ramona Park is a junior at Santa Fe High School who holds interests in graphic design and the production of creative content. She is aimed at pursuing a career related to journalism or computer science. During this project although Ramona did not take a major role in the coding, she contributed to research, writing, and final edit of the paper. In addition to her participation in the challenge as a team member, she also submitted an original logo design for the t-shirt competition.

email: yoharamona@gmail.com

References

[1] Nicholson, Christie. "Pollution-Killing Material: Smog-Eating Cement." Core77. N.p., 8 Nov. 2015. Web. 27 Mar. 2016.

[2] Kaplan, Sarah. "Air Pollution in China Is Killing 1.6 Million People a Year, Researchers Say." Washington Post. The Washington Post, 14 Aug. 2015. Web. 05 Apr. 2016.

[3] Renowden, John, CEnf, FIMechE. Smog-Eating Tile: A Real-World Product for Reducing the Harmful Health Effects of Contaminated Air. Interface, Jan. 2012. PDF.

[4] Albuquerque Air Quality Improved Finds 2016 'State of the Air' Report." American Lung Association. American Lung Association, n.d. Web. 02 Apr. 2017.

[5] Williams, Ed. "Neighborhood Holds Breath for Better Air Quality." Publichealthnm.org. Public Health New Mexico, 26 May 2016. Web. 02 Apr. 2017.

[6] Silko, John R. "AIR POLLUTION IN ALBUQUERQUE— BERNALILLO COUNTY." Natural Resources Journal (1970): n. pag. Web. 02 Apr. 2017.

[7] Rinkesh. "Causes and Effects of Smog - Conserve Energy Future." ConserveEnergyFuture. N.p., 22 Feb. 2013. Web. 3 Nov. 2015.

[8] "Smog." National Geographic Education. National Geographic Society, 21 Jan. 2011. Web. 3 Nov. 2015.

[9] Zhang, Kai, Stuart Batterman, and François Dion. "Vehicle Emissions in Congestion: Comparison of Work Zone, Rush Hour and Free-flow Conditions." Atmospheric Environment 45.11 (2011): 1929-939. 14 Jan. 2011. Web. 2 Apr. 2017.

[10] Zhang, Kai, and Stuart Batterman. "Air Pollution and Health Risks Due to Vehicle Traffic." The Science of the Total Environment. U.S. National Library of Medicine, 15 Apr. 2013. Web. 02 Apr. 2017.

[11] Gurol, Mirat D. Photo-Catalytic Construction Materials and Reduction in Air Pollutants. Sacramento: California State University Faculty Research Fellows Program, Mar. 2006. PDF.

[12] Boonen, Elia, and Anne Beeldens. "Photocatalytic Roads: From Lab Tests to Real Scale Applications." SpringerLink. Springer Berlin Heidelberg, 09 Sept. 2012. Web. 03 Apr. 2017.

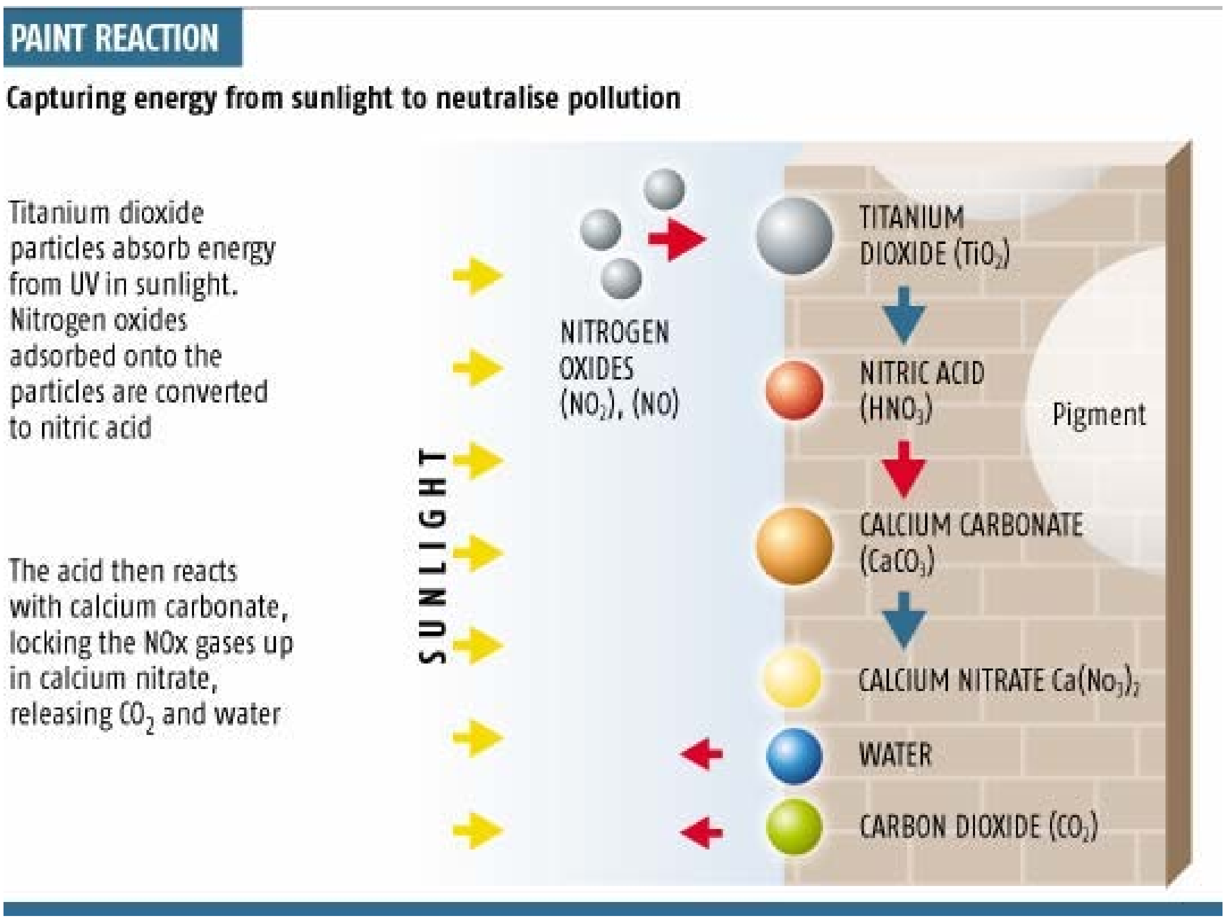
[13] Heidorn, Keith C., PhD. "Winds of the City." Weather Phenomenon and Elements. 1 Nov. 2005. Web. 15 Feb. 2016.

[14] Spirn, Anna Whiston. "Urban Street Canyons." Urban Street Canyons. M.I.T, n.d. Web. 03 Apr. 2017.

[15] Berdahl, Paul, and Akbari Hashem. Evaluation of Titanium Dioxide as a Photocatalyst for Removing Air Pollutants. Rep. no. CEC-500-2007-112. California Energy Commission, Pier Energy-Related Environmental Research Program. Print.

[16] Cahill, Rowan, Lisel Faust, Theo Goujon, and Ramona Park. *What is the Best Placement to Maximize the Effectiveness of Smog-Reducing Materials?* Rep. Los Alamos: SuperComputing Challenge, 2016. Print.

Appendix A: TiO_2 Diagram



Hogan, Jenny. *Paint Reaction*. Digital image. *New Scientist*. Reed Business Information Ltd., 4 Feb. 2004. Web. 29 Mar. 2016.

Appendix B: Real World Applications

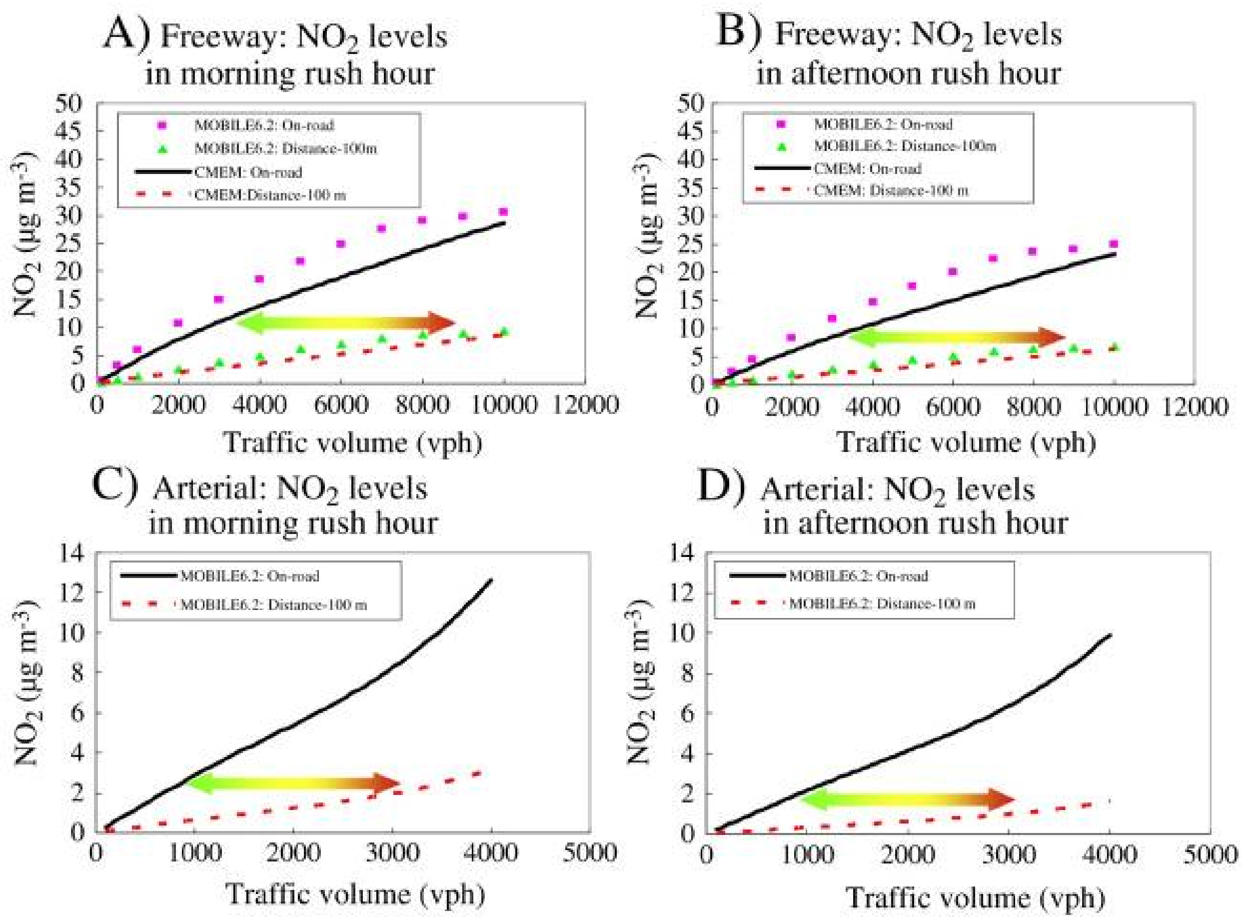
The Palazzo Italia in Milan, Italy
Source: Nemesi & Partners

The Manuel Gea González Hospital in Mexico
Source: Elegant Embellishments

TX Active cement, Italcementi R&D center
Source: Liao Yusheng

Pavers in Hengelo, Netherlands; 25-45% reduction
Source: Science Direct

Appendix C: Freeway and Arterial NO_2 Comparisons Graph [6]



Appendix D: Raw Data Averages and Total Percentage Processed

Ticks	None	Bottom	Middle	Top	Full
1	238059	238059	238059	238059	238059
2	442806.8	430572.4	439465.5	444636	433498.5
3	854384.6	829548	848947.5	850290.8	823503.4
4	1240451	1197858	1219769	1222678	1163211
5	1483576	1433403	1454301	1460791	1376207
6	1651880	1599818	1618026	1623014	1527363
7	1788828	1734588	1748827	1754079	1653993
8	1914266	1857529	1870206	1873830	1766448
9	2035716	1976805	1989421	1990113	1875623
10	2156571	2093860	2106533	2100519	1981331
11	2268500	2203814	2213868	2209322	2084657
12	2387438	2319415	2325578	2320404	2190773
13	2503258	2433156	2438913	2429612	2296904
14	2610687	2538705	2543684	2532830	2395396
15	2722913	2647654	2651403	2637654	2494476
16	2833333	2755556	2759211	2744378	2594042
17	2940341	2860019	2860981	2846050	2691272
18	3045543	2966433	2964354	2946324	2787828
19	3149847	3068452	3066768	3044014	2880532
20	3254661	3166844	3167346	3143242	2974647
21	3356994	3266970	3263723	3239372	3066771
22	3459689	3371092	3363312	3338800	3160056
23	3561084	3468248	3460874	3433691	3252211
24	3656721	3564618	3555016	3524445	3339772
25	3756382	3659438	3649114	3615698	3428011
26	3853015	3755052	3743405	3708686	3514820
27	3948122	3850085	3837800	3800324	3605300
28	4045025	3943289	3929708	3891503	3689854
29	4135384	4031866	4018143	3977886	3771917
30	4228534	4121958	4107152	4063379	3855438
31	4322660	4214676	4196613	4153673	3941703
32	4414514	4305491	4288953	4239071	4025241
33	4502075	4393329	4374213	4324114	4104469
34	4590776	4479654	4459330	4407436	4185736
35	4677740	4565633	4544843	4490593	4265008
36	4765787	4646747	4627567	4563591	4327361
37	4852913	4715597	4706694	4637142	4374776
38	4939900	4800681	4787114	4716198	4449249

39	5022773	4880517	4869518	4794613	4525919
40	5105042	4962819	4950902	4873807	4601846
41	5189160	5046850	5033117	4953596	4677305
42	5273006	5128002	5113550	5032945	4754039
43	5354215	5207396	5191202	5109443	4827367
44	5434638	5287481	5268571	5182694	4900696
45	5514001	5366301	5346010	5259468	4972982
46	5594096	5443851	5423399	5332507	5045959
47	5671499	5522317	5499569	5408876	5116807
48	5749687	5596614	5575338	5480605	5188091
49	5826070	5672869	5648231	5555216	5257356
50	5902001	5747527	5722120	5627448	5326577
51	5977338	5821234	5793896	5698272	5395093
52	6052013	5895683	5867100	5767770	5463943
53	6127316	5969541	5939818	5839894	5531038
54	6199055	6042134	6010956	5907968	5598086
55	6271935	6112935	6080792	5975940	5664475
56	6343751	6183111	6150067	6042715	5730372
57	6414326	6253801	6221354	6110687	5795280
58	6486602	6322408	6288783	6178143	5859361
59	6556073	6392260	6357612	6244293	5922102
60	6623694	6460449	6424066	6309664	5987932
61	6692493	6526821	6489146	6374538	6049485
62	6759764	6594267	6557115	6438798	6110346
63	6827807	6661828	6622712	6502478	6171825
64	6894818	6727853	6688535	6565870	6232363
65	6959972	6792822	6751478	6627690	6293238
66	7026467	6857642	6816143	6690293	6353974

Total Percentage of Smog Processed

None	Bottom	Middle	Top	Full
0	2.402691	2.9933	4.784391	9.570847

Appendix D: Code

```
Extensions [
    profiler
]

Patches-own [
    Surface TiO2
    w-direction w-speed
    s-close
    smog nsmog
    tempsmog
    Restriction
    strue tcount
    s-temp d-temp
]

Globals [
    osmog scount
]

Breed [
    sun
]

to Setup
    clear-all
    reset-ticks
    ask patches [
        set restriction (count neighbors6 with [surface = true])
    ]
    run [
        Roads
        Buildings
        Placement
        Wind-set
    ]
    repeat world-width [
        run [
            Direction-set
        ]
    ]
    run [
        sclose
```

```
    ]
    ask patches [
        if count neighbors4 with [pcolor = grey and surface = true] = 4 and pzcor = 0 [
            ask patch [pxcor] of self [pycor] of self ([pzcor] of self + 1) [
                set smog 100
            ]
        ]
    ]
end

to Simulate
    run [
        Sunlight
        Diffusion
        Removal
    ]
    if ticks = 66 [
        stop
    ]
end

to Roads
    ask patches [
        if pzcor = 0 [
            set pcolor green
            set surface true
        ]
        if pzcor = 0 and pycor >= -1 and pycor <= 1 [
            set pcolor grey
        ]
        if pzcor = 0 and pycor >= 39 and pycor <= 41 [
            set pcolor grey
        ]
        if pzcor = 0 and pycor <= -39 and pycor >= -41 [
            set pcolor grey
        ]
        if pzcor = 0 and (pycor = 79 or pycor = 80 or pycor = -79 or pycor = -80) [
            set pcolor grey
        ]
        if pzcor = 0 and pxcor >= -1 and pxcor <= 1 [
            set pcolor grey
        ]
        if pzcor = 0 and pxcor >= 37 and pxcor <= 39 [
            set pcolor grey
```

```
        ]
        if pzcor = 0 and pxcor <= -37 and pxcor >= -39 [
            set pcolor grey
        ]
        if pzcor = 0 and (pxcor = 75 or pxcor = 76 or pxcor = -75 or pxcor = -76) [
            set pcolor grey
        ]
    ]
end

to Buildings
    ask patches [
        if pzcor <= 22 [
            if (pxcor > 20 and pxcor < 36) and (pycor > 21 and pycor < 38) [
                set pcolor grey
                set surface true
            ]
        ]
        if pzcor <= 18 [
            if (pxcor > 2 and pxcor < 19) and (pycor > 2 and pycor < 20) [
                set pcolor grey
                set surface true
            ]
        ]
        if pzcor <= 14 [
            if (pxcor > -36 and pxcor < -20) and (pycor > -78 and pycor < -42) [
                set pcolor grey
                set surface true
            ]
        ]
        if pzcor <= 12 [
            if (pxcor > -25 and pxcor < -14) and (pycor > -38 and pycor < -2) [
                set pcolor grey
                set surface true
            ]
        ]
        if pzcor <= 10 [
            if (pxcor > 40 and pxcor < 57) and (pycor > 61 and pycor < 78) [
                set pcolor grey
                set surface true
            ]
        ]
        if pzcor <= 8 [
            if (pxcor > -36 and pxcor < -25) and (pycor > -38 and pycor < -2) [
```

```
            set pcolor grey
            set surface true
        ]
        if (pxcor > 58 and pxcor < 74) and (pycor > 42 and pycor < 60) [
            set pcolor grey
            set surface true
        ]
    ]
    if pzcor <= 7 [
        if (pxcor > 40 and pxcor < 57) and (pycor > 42 and pycor < 60) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 2 and pxcor < 36) and (pycor >= -20 and pycor < -2) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 2 and pxcor < 19) and (pycor > -38 and pycor <= -21) [
            set pcolor grey
            set surface true
        ]
    ]
    if pzcor <= 6 [
        if (pxcor > 40 and pxcor < 57) and (pycor > 2 and pycor < 38) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 2 and pxcor <= 13) and (pycor > 42 and pycor < 78) [
            set pcolor grey
            set surface true
        ]
        if (pxcor >= 14 and pxcor <= 24) and (pycor > 42 and pycor < 60) [
            set pcolor grey
            set surface true
        ]
        if (pxcor >= 25 and pxcor < 36) and (pycor > 42 and pycor < 78) [
            set pcolor grey
            set surface true
        ]
    ]
    if pzcor <= 5 [
        if (pxcor > -14 and pxcor < -2) and (pycor > -38 and pycor < -2) [
            set pcolor grey
            set surface true
```

```
        ]
        if (pxcor > 40 and pxcor < 57) and (pycor > -38 and pycor < -2) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 40 and pxcor < 57) and (pycor > -60 and pycor < -42) [
            set pcolor grey
            set surface true
        ]
    ]
    if pzcor <= 4 [
        if (pxcor > -36 and pxcor < -2) and (pycor > 42 and pycor < 78) [
            set pcolor grey
            set surface true
        ]
    ]
    if pzcor <= 3 [
        if (pxcor > 20 and pxcor < 36) and (pycor > -78 and pycor < -42) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 58 and pxcor < 74) and (pycor > -60 and pycor < -42) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 58 and pxcor < 74) and (pycor > 2 and pycor < 20) [
            set pcolor grey
            set surface true
        ]
    ]
    if pzcor <= 2 [
        if (pxcor > -19 and pxcor < -2) and (pycor > -60 and pycor < -42) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 58 and pxcor < 74) and (pycor > 21 and pycor < 38) [
            set pcolor grey
            set surface true
        ]
        if (pxcor > 58 and pxcor < 74) and (pycor > -38 and pycor < -2) [
            set pcolor grey
            set surface true
        ]
    ]
```

```
    ]
end

to Placement
    ask patches [
        if Material = "Bottom" [
            if pzcor <= 2 and pzcor > 0 [
                set pcolor white
                set TiO2 true
            ]
        ]
        if Material = "Middle" [
            if pzcor > 2 and pzcor <= 4 [
                set pcolor white
                set TiO2 true
            ]
        ]
        if Material = "Top" [
            if pzcor > 4 and pzcor <= 22 [
                set pcolor white
                set TiO2 true
            ]
        ]
        if Material = "Full" [
            if pzcor <= 22 and pzcor != 0 [
                set pcolor white
                set TiO2 true
            ]
        ]
    ]
end

To Wind-set
    if Direction-O = "South" [
        ask patches with [pycor = min-pycor and surface != true] [
            set w-direction 3
        ]
    ]
    if Direction-O = "North" [
        ask patches with [pycor = max-pycor and surface != true] [
            set w-direction 4
        ]
    ]
    if Direction-O = "West" [
```

```
        ask patches with [pxcor = min-pxcor and surface != true] [
            set w-direction 2
        ]
    ]
    if Direction-O = "East" [
        ask patches with [pxcor = max-pxcor and surface != true] [
            set w-direction 5
        ]
    ]
end

to Direction-set
    ask patches with [surface != true]  [
        if Direction-O = "South" [
            if w-direction = 3 [
                ask patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self  [
                    if surface != true  [
                        set w-direction 3
                    ]
                    if [surface] of (patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self) = true [
                        set w-direction 6
                    ]
                ]
            ]
            if [surface] of (patch ([pxcor] of self - 1) ([pycor] of self - 1) [pzcor] of self) = true and
            (count neighbors4 with [surface = true]) = 0 [
                ask patch ([pxcor] of self - 1) ([pycor] of self) [pzcor] of self [
                    set w-direction 5
                ]
            ]
            if [surface] of (patch ([pxcor] of self + 1) ([pycor] of self - 1) [pzcor] of self) = true and
            (count neighbors4 with [surface = true]) = 0 [
                ask patch ([pxcor] of self + 1) ([pycor] of self) [pzcor] of self [
                    set w-direction 2
                ]
            ]
            if w-direction = 6 [
                ask patch [pxcor] of self [pycor] of self ([pzcor] of self - 1) [
                    if surface != true [
                        set w-direction 6
                    ]
                ]
            ]
        ]
```

```
if Direction-O = "North" [
    if w-direction = 4 [
        ask patch [pxcor] of self ([pycor] of self - 1) [pzcor] of self [
            if surface != true [
                set w-direction 4
            ]
            if [surface] of (patch [pxcor] of self ([pycor] of self - 1) [pzcor] of self) = true [
                set w-direction 6
            ]
        ]
    ]
    if [surface] of (patch ([pxcor] of self - 1) ([pycor] of self + 1) [pzcor] of self) = true and
    (count neighbors4 with [surface = true]) = 0 [
        ask patch ([pxcor] of self - 1) ([pycor] of self) [pzcor] of self [
            set w-direction 5
        ]
    ]
    if [surface] of (patch ([pxcor] of self + 1) ([pycor] of self + 1) [pzcor] of self) = true and
    (count neighbors4 with [surface = true]) = 0 [
        ask patch ([pxcor] of self + 1) ([pycor] of self) [pzcor] of self [
            set w-direction 2
        ]
    ]
    if w-direction = 6 [
        ask patch [pxcor] of self [pycor] of self ([pzcor] of self - 1) [
            if surface != true [
                set w-direction 6
            ]
        ]
    ]
]
if Direction-O = "West" [
    if w-direction = 2 [
        ask patch ([pxcor] of self + 1) [pycor] of self [pzcor] of self [
            if surface != true [
                set w-direction 2
            ]
            if [surface] of (patch ([pxcor] of self + 1) [pycor] of self [pzcor] of self) = true [
                set w-direction 6
            ]
        ]
    ]
    if [surface] of (patch ([pxcor] of self - 1) ([pycor] of self - 1) [pzcor] of self) = true and
    (count neighbors4 with [surface = true]) = 0 [
```

```
            ask patch ([pxcor] of self) ([pycor] of self - 1) [pzcor] of self [
                set w-direction 4
            ]
        ]
        if [surface] of (patch ([pxcor] of self - 1) ([pycor] of self + 1) [pzcor] of self) = true and
        (count neighbors4 with [surface = true]) = 0 [
            ask patch ([pxcor] of self) ([pycor] of self + 1) [pzcor] of self [
                set w-direction 3
            ]
        ]
        if w-direction = 6 [
            ask patch [pxcor] of self [pycor] of self ([pzcor] of self - 1) [
                if surface != true [
                    set w-direction 6
                ]
            ]
        ]
    ]
    if Direction-O = "East" [
        if w-direction = 5 [
            ask patch ([pxcor] of self - 1) [pycor] of self [pzcor] of self [
                if surface != true [
                    set w-direction 5
                ]
                if [surface] of (patch ([pxcor] of self - 1) [pycor] of self [pzcor] of self) = true [
                    set w-direction 6
                ]
            ]
        ]
        if [surface] of (patch ([pxcor] of self + 1) ([pycor] of self - 1) [pzcor] of self) = true and
        (count neighbors4 with [surface = true]) = 0 [
            ask patch ([pxcor] of self) ([pycor] of self - 1) [pzcor] of self [
                set w-direction 4
            ]
        ]
        if [surface] of (patch ([pxcor] of self + 1) ([pycor] of self + 1) [pzcor] of self) = true and
        (count neighbors4 with [surface = true]) = 0 [
            ask patch ([pxcor] of self) ([pycor] of self + 1) [pzcor] of self [
                set w-direction 3
            ]
        ]
        if w-direction = 6 [
            ask patch [pxcor] of self [pycor] of self ([pzcor] of self - 1) [
                if surface != true [
```

```
                    set w-direction 6
                ]
            ]
        ]
    ]
  ]
End

to Sclose
  if Direction-O = "North" or Direction-O = "South" [
    ask patches with [surface = true] [
      if [surface] of patch ([pxcor] of self - 6) [pycor] of self [pzcor] of self = true [
        ask patch ([pxcor] of self - 1) [pycor] of self [pzcor] of self [
          if surface != true [
            set s-close true
          ]
        ]
        ask patch ([pxcor] of self - 2) [pycor] of self [pzcor] of self [
          if surface != true [
            set s-close true
          ]
        ]
        ask patch ([pxcor] of self - 3) [pycor] of self [pzcor] of self [
          if surface != true [
            set s-close true
          ]
        ]
      ]
      if [surface] of patch ([pxcor] of self + 6) [pycor] of self [pzcor] of self = true [
        ask patch ([pxcor] of self + 1) [pycor] of self [pzcor] of self [
          if surface != true [
            set s-close true
          ]
        ]
        ask patch ([pxcor] of self + 2) [pycor] of self [pzcor] of self [
          if surface != true [
            set s-close true
          ]
        ]
        ask patch ([pxcor] of self + 3) [pycor] of self [pzcor] of self [
          if surface != true [
            set s-close true
          ]
        ]
```

```
        ]
    ]
    ask patches with [s-close = true] [
        if Direction-O = "North" [
            set s-temp 0
            while [s-temp != 5] [
                set s-temp s-temp + 1
                ask patch [pxcor] of self ([pycor] of self - s-temp) [pzcor] of self [
                    set s-close true
                ]
            ]
        ]
    ]
    ask patches with [s-close = true] [
        if Direction-O = "South" [
            set s-temp 0
            while [s-temp != 5] [
                set s-temp s-temp + 1
                ask patch [pxcor] of self ([pycor] of self + s-temp) [pzcor] of self [
                    set s-close true
                ]
            ]
        ]
    ]
]
if Direction-O = "East" or Direction-O = "West" [
    ask patches with [Surface = true] [
        if [surface] of patch [pxcor] of self ([pycor] of self - 6) [pzcor] of self = true [
            ask patch [pxcor] of self ([pycor] of self - 1) [pzcor] of self [
                if surface != true [
                    set s-close true
                ]
            ]
            ask patch [pxcor] of self ([pycor] of self - 2) [pzcor] of self [
                if surface != true [
                    set s-close true
                ]
            ]
            ask patch [pxcor] of self ([pycor] of self - 3) [pzcor] of self [
                if surface != true [
                    set s-close true
                ]
            ]
        ]
```

```
            if [surface] of patch [pxcor] of self ([pycor] of self + 6) [pzcor] of self = true [
                ask patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self [
                    if surface != true [
                        set s-close true
                    ]
                ]
                ask patch [pxcor] of self ([pycor] of self + 2) [pzcor] of self [
                    if surface != true [
                        set s-close true
                    ]
                ]
                ask patch [pxcor] of self ([pycor] of self + 3) [pzcor] of self [
                    if surface != true [
                        set s-close true
                    ]
                ]
            ]
        ]
        ask patches with [s-close = true] [
            if Direction-O = "West" [
                set s-temp 0
                while [s-temp != 5] [
                    set s-temp s-temp + 1
                    ask patch ([pxcor] of self + s-temp) [pycor] of self [pzcor] of self [
                        set s-close true
                    ]
                ]
            ]
        ]
        ask patches with [s-close = true] [
            if Direction-O = "East" [
                set s-temp 0
                while [s-temp != 5] [
                    set s-temp s-temp + 1
                    ask patch ([pxcor] of self - s-temp) [pycor] of self [pzcor] of self [
                        set s-close true
                    ]
                ]
            ]
        ]
    ]
end

to Reset
```

```
    ask patches with [pzcor = 0] [
        set strue false
    ]
    ask patches with [pycor > -8 and pycor < -4 and pzcor < 8] [
        set strue false
    ]
    ask patches with [pycor > -2 and pycor < 2 and pzcor < 8] [
        set strue false
    ]
end

to Diffusion
    run [
        Diffusion-E
        Diffusion-WT-2
        Diffusion-WT-5
        Diffusion-WT-3
        Diffusion-WT-4
        Clear
    ]
    tick
end

To Diffusion-E
    ask Patches with [s-close != true and w-direction != 6] [
        if smog > 0.0001 [
            set osmog smog
            ask neighbors6 [
                if surface != true and smog <= 0.0001 [
                    set smog (osmog / (7 - restriction))
                    if smog > .0001 [
                        set pcolor (37 - (smog / 20))
                    ]
                ]
                if smog > 0.0001 [
                    set nsmog smog
                    set smog (nsmog + (osmog / (7 - restriction)))
                    if smog > 100 [
                        set smog 100
                    ]
                    if smog > .0001 [
                        set pcolor (37 - (smog / 20))
                    ]
                ]
```

```
            ]
            set smog (osmog / (7 - restriction))
        ]
         if smog > .0001 [
            set pcolor (37 - (smog / 20))
        ]
    ]
end

to Diffusion-WT-2
    set scount 0
    set osmog count patches with [smog > .0001]
    ask patches with [s-close = true and w-direction = 2] [
        if smog > .0001 [
            ask patch ([pxcor] of self + 1) [pycor] of self [pzcor] of self [
                set tempsmog [smog] of patch ([pxcor] of self - 1) [pycor] of self [pzcor] of self
                set scount scount + 1
            ]
            set smog 0
        ]
    ]
    if osmog = scount [
        ask patches with [tempsmog > 0 and s-close = true and w-direction = 2] [
            set smog tempsmog
            set tempsmog 0
            if smog > .0001 [
                set pcolor (37 - (smog / 20))
            ]
        ]
    ]
    ask patches with [surface != true and pxcor != max-pxcor and smog > .0001 and s-close =
    true and w-direction = 2] [
        set osmog smog
        ask neighbors6 [
            set nsmog smog
            if surface != true and smog < 0.0001 [
                set smog (osmog / (6 - restriction))
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
```

```
                ]
            ]
            if surface != true and smog > 0.0001 [
                set smog (nsmog + (osmog / (6 - restriction)))
                if smog > 100 [
                    set smog 100
                ]
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
        ]
        ask patch ([pxcor] of self - 1) [pycor] of self [pzcor] of self [
            set smog nsmog
        ]
        set smog (osmog / (6 - restriction))
    ]
end

to Diffusion-WT-5
    set scount 0
    set osmog count patches with [smog > .0001]
    ask patches with [s-close = true and w-direction = 5] [
        if smog > .0001 [
            ask patch ([pxcor] of self - 1) [pycor] of self [pzcor] of self [
                set tempsmog [smog] of patch ([pxcor] of self + 1) [pycor] of self [pzcor] of self
                set scount scount + 1
            ]
            set smog 0
        ]
    ]
    if osmog = scount [
        ask patches with [tempsmog > 0 and s-close = true and w-direction = 5] [
            set smog tempsmog
            set tempsmog 0
            if smog > .0001 [
                set pcolor (37 - (smog / 20))
            ]
        ]
    ]
```

```
    ask patches with [surface != true and pxcor != min-pxcor and smog > .0001 and s-close =
    true and w-direction = 5] [
        set osmog smog
        ask neighbors6 [
            set nsmog smog
            if surface != true and smog < 0.0001 [
                set smog (osmog / (6 - restriction))
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
            if surface != true and smog > 0.0001 [
                set smog (nsmog + (osmog / (6 - restriction)))
                if smog > 100 [
                    set smog 100
                ]
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
        ]
        ask patch ([pxcor] of self + 1) [pycor] of self [pzcor] of self [
            set smog nsmog
        ]
        set smog (osmog / (6 - restriction))
    ]
end

to Diffusion-WT-3
    set scount 0
    set osmog count patches with [smog > .0001]
    ask patches with [s-close = true and w-direction = 3] [
        if smog > .0001 [
            ask patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self [
                set tempsmog [smog] of patch [pxcor] of self ([pycor] of self - 1) [pzcor] of self
                set scount scount + 1
            ]
            set smog 0
        ]
    ]
    if osmog = scount [
        ask patches with [tempsmog > 0 and s-close = true and w-direction = 3] [
            set smog tempsmog
            set tempsmog 0
```

```
        if smog > .0001 [
          set pcolor (37 - (smog / 20))
        ]
      ]
    ]
    ask patches with [surface != true and pycor != max-pxcor and smog > .0001 and s-close =
    true and w-direction = 3] [
      set osmog smog
      ask neighbors6 [
        set nsmog smog
        if surface != true and smog < 0.0001 [
          set smog (osmog / (6 - restriction))
          if smog > .0001 [
            set pcolor (37 - (smog / 20))
          ]
        ]
        if surface != true and smog > 0.0001 [
          set smog (nsmog + (osmog / (6 - restriction)))
          if smog > 100 [
            set smog 100
          ]
          if smog > .0001 [
            set pcolor (37 - (smog / 20))
          ]
        ]
      ]
      ask patch [pxcor] of self ([pycor] of self - 1) [pzcor] of self [
        set smog nsmog
      ]
      set smog (osmog / (6 - restriction))
    ]
end

to Diffusion-WT-4
  set scount 0
  set osmog count patches with [smog > .0001]
  ask patches with [s-close = true and w-direction = 4] [
    if smog > .0001 [
      ask patch [pxcor] of self ([pycor] of self - 1) [pzcor] of self [
        set tempsmog [smog] of patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self
        set scount scount + 1
      ]
      set smog 0
    ]
```

```
    ]
    if osmog = scount [
        ask patches with [tempsmog > 0 and s-close = true and w-direction = 4] [
            set smog tempsmog
            set tempsmog 0
            if smog > .0001 [
                set pcolor (37 - (smog / 20))
            ]
        ]
    ]
    ask patches with [surface != true and pycor != min-pxcor and smog > .0001 and s-close =
    true and w-direction = 4] [
        set osmog smog
        ask neighbors6 [
            set nsmog smog
            if surface != true and smog < 0.0001 [
                set smog (osmog / (6 - restriction))
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
            if surface != true and smog > 0.0001 [
                set smog (nsmog + (osmog / (6 - restriction)))
                if smog > 100 [
                    set smog 100
                ]
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
        ]
        ask patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self [
            set smog nsmog
        ]
        set smog (osmog / (6 - restriction))
    ]
end

to Diffusion-WT-6
    set scount 0
    set osmog count patches with [smog > .0001]
    ask patches with [w-direction = 6] [
        if smog > .0001 [
            ask patch [pxcor] of self [pycor] of self ([pzcor] of self - 1) [
```

```
                set tempsmog [smog] of patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self
                set scount scount + 1
            ]
            set smog 0
        ]
    ]
    if osmog = scount [
        ask patches with [tempsmog > 0 and s-close = true and w-direction = 6] [
            set smog tempsmog
            set tempsmog 0
            if smog > .0001 [
                set pcolor (37 - (smog / 20))
            ]
        ]
    ]
    ask patches with [surface != true and smog > .0001 and w-direction = 6] [
        set osmog smog
        ask neighbors6 [
            set nsmog smog
            if surface != true and smog < 0.0001 [
                set smog (osmog / (6 - restriction))
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
            if surface != true and smog > 0.0001 [
                set smog (nsmog + (osmog / (6 - restriction)))
                if smog > 100 [
                    set smog 100
                ]
                if smog > .0001 [
                    set pcolor (37 - (smog / 20))
                ]
            ]
        ]
        ask patch [pxcor] of self ([pycor] of self + 1) [pzcor] of self [
            set smog nsmog
        ]
        set smog (osmog / (6 - restriction))
    ]
end

to Clear
    ask patches [
```

```
            if surface != true and smog < .0001 [
                set pcolor [0 0 0 0]
                set smog 0
            ]
            if pxcor = max-pxcor or pxcor = min-pxcor or pycor = max-pycor or pycor = min-pycor or
            pzcor = max-pzcor [
                set smog 0
            ]
            if count neighbors4 with [pcolor = grey and surface = true] = 4 and pzcor = 0 [
                ask patch [pxcor] of self [pycor] of self ([pzcor] of self + 1) [
                    ifelse ticks < 21 or ticks > 55 [
                        set smog 100
                    ]
                    [
                        set smog 25
                    ]
                ]
            ]
        ]
end

To Removal
        ask patches [
            if TiO2 = true and strue = true [
               ask neighbors6 [
                    set nsmog smog
                    if smog > 0 [
                        set smog 0
                    ]
                ]
            ]
            if TiO2 = true and strue != true [
                ask neighbors6 [
                    set nsmog smog
                    if smog > 0 [
                        set smog (.5 * nsmog)
                    ]
                ]
            ]
        ]
end
```