Gas Diffusion

Category A
New Mexico Supercomputing Challenge
Final Report
April 1, 2001

Team:51
Picacho Middle School

Team Members:
Jordan Aday
Hector Cardona
Eddie Banda

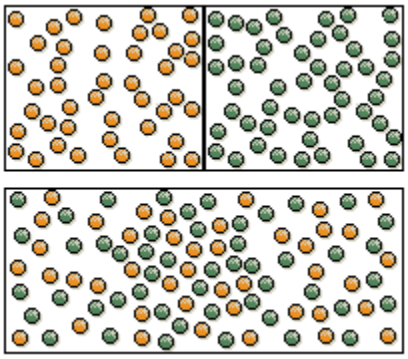
Teacher:
Jean McCray

Mentor:
Shaun Cooper

Table of Contents

Executive Summary

Gas diffusion is everywhere. Due to numerous natural gas leaks at Picacho Middle School our team chose to understand how gas diffuses. Our methodologies were calculated numerically and then modeled in two phases with C programming. The first phase of programming confirmed our numerical calculations. The second phase allowed us to randomly distribute incoming gases. The results of the second phase could not confirm our numerical calculations because the random generator used different inputs, however, we were able to understand diffusion in a more mathematically complex situation. Our models did not include time as a factor and so a conclusion about the safety levels at Picacho Middle School during these leaks could not be drawn.



Diffusion- the flow of energy or matter from a higher concentration to a lower concentration, resulting in a homogeneous distribution. Diffusion occurs most rapidly in gasses but all types of diffusion follow the same laws. The rate of diffusion is proportional to the cross sectional area and to the gradient of concentration, temperature or change. Diffusion is not mixing. Instead, it is a molecular process, depending solely on the random motions of individual molecules.

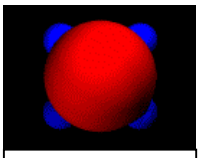
Introduction

Diffusion is everywhere! Diffusion is a natural phenomenon that we all experience in our everyday lives. It is in the weather. It happens with light. It is when oil seeps into the ground and contaminates ground water, it takes place in our digestive system, and it occurs when you open a bottle of perfume and the smell seeps into the air. After a few minutes, someone at the other end of the room can smell the perfume. What happens in all cases of diffusion is this: the molecules migrate from the most concentrated area to ones that are less concentrated.

Diffusion is part of our lives and only a few of us are aware of it ever being there. The reason gas diffusion interested us and we chose this topic for our supercomputing project was because we have had numerous gas leaks at Picacho Middle School over the last two years. The school had to be evacuated until the leaks were contained. We wanted to try and figure out, how the gas was actually diffusing

throughout the school. After completing our research, we discovered that a “third party”, not the owner of the line, causes more than half of natural gas incidences. The “third party” at Picacho Middle School was a construction company on all occasions that were working around, and in our school.

Natural gas, as we know it consists of 80-90% methane. Methane (also known as CH4) is a colorless odorless gas that is extremely flammable. Short-term effects of breathing methane are vomiting, nausea, dizziness. These can lead to a coma and even death.



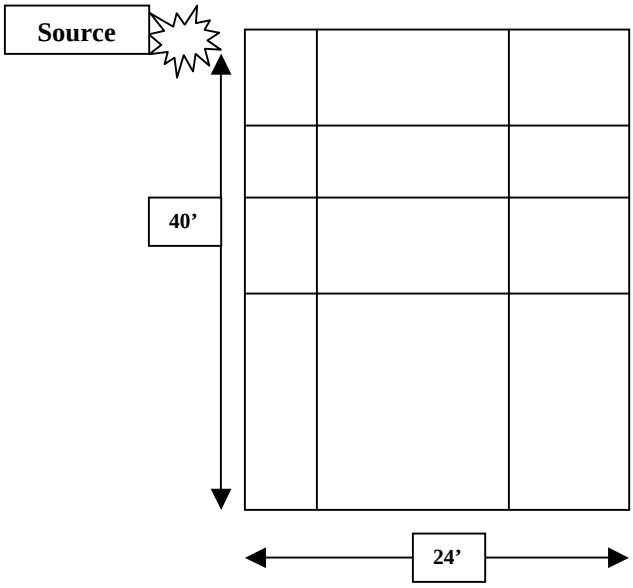
Ch4 molecule

Description

Graham’s Law of Diffusion states that the rate of diffusion of a gas is inversely proportional to the square root of its density provided that its temperature and pressure remains unchanged.

The rate of diffusion of the gases depends on the density of the gas. It also depends on the molecular weight of the gas, since the density of the gas increases as its molecular weight increases. The density of methane, the gas we are using in our model is .717 g/l (grams per liter) at O degrees Celsius. Due to limited resources and the complex algebraic math problems involved, we relied on a simple model to demonstrate gas diffusion for our project. It was beyond our mathematical skills at this time to address the diffusion formulas associated with gas diffusion. We completed a 2-Dimensional C program that is a simple model. This program is approximates with a persistent injection source. The dispersion is based on a per of unit time, with the time unit, being unimportant. A new “pulse” does not happen until the first pulse of gas has been completely diffused throughout the room.

We picked our technology classroom at Picacho Middle School to use as our testing area. The room is 24’x 40’. We divided the room into 8’x 8’ grids so our model looked like the grid below, with the gas source in the upper left hand corner.



The first model we made, showed diffusion in 2 directions. The 2 directions we chose were right and down. One hundred “molecules” entered the room each “pulse” in the upper left hand corner and that would diffuse throughout the room before the next “pulse” of gas would enter. The model diffused 40% of the molecules in each way. Meaning 40% of gas in that grid would go down and 40% would go left.

For the first phase of the project we first did the calculations by hand (Appendix I) and then wrote a C program that confirmed our results were correct. (Appendix II) We completed a C program that diffused gas in this manner for 5 “pulses”. Each pulse, added another 100 molecules in the upper left corner and added to the molecules that were left in the grid from the previous “pulse”, each time 40% of the molecules in each grid diffused to the right, 40% down.

Even though this left us with a simple example of the rate of diffusion after five pulses it proved our code could work with the inputted percents and easily calculate the impulses. Once we had a working code we then modified the existing program and changed the process by using random percentages in the grids. This was useful because it began to insert changes in direction like walls and wind that would interfere with the natural rate of diffusion in a “real” situation.

For the second phase of our project, again, we first did our legwork by hand calculations (Appendix III), and used the same room as our test area. This grid differed in only one way. Instead of the gas “leaving the room”, it changed direction when it hit the wall. For our hand calculations we used different percentages for each column of grids. As soon as we did the hand calculations, we modified our existing C code to write a program that would use random numbers and the gas would now go in four directions, instead of two. The directions in the second phase of our 2-Dimensional model were up, down, right and left.

For the second phase of our program we used “Rand_Max” in our code, which randomly picks numbers (percentages) to disperse the gas throughout the room. This is a much better program then the one in our first phase. One problem with this is that we cannot reproduce the same results more than once, and it also does not display the numbers (percentages) that it chose to use for diffusion.

Results

The results of our program were that we were able to see the rate at which the natural gas diffused into the air, however, the results are only in figures and not in a visual aid that could show the data in a more user friendly way. In both the first and second phases of our project the results of our program were what we were striving for; fact that it computed a simple model of gas diffusion and it showed that data in numerical form. In the beginning we tried to see if we were safe at Picacho Middle School during one of these gas leaks. After completing our programming and our research, we were unable to confirm if we were actually safe. The only safety factors we found were the combustible limits of natural gas in air. The combustible limits are the lowest and the highest concentrations of a specific gas in mixing with air that can be ignited at ordinary temperature and pressure of the mixture. The combustible limits of natural gas with air are 5-15%. In our program, we can figure out which “grid” contains these

limits, however, at this time we did not figure time in our problem. Time is irrelevant in our program. Without this factor it is impossible to determine safety levels.

Conclusions

We wrote code to make a simple model of diffusion. Our first goal was to get the program to work. After achieving this, we tried to take the problem even farther but ran out of time. Our results were what we expected in the first phase of our project because we first hand calculated our results. In the second phase of our project, we used random numbers so each time we ran the program we would get different data. We were surprised at the second phase of our project because the percentages were much smaller when we used random numbers.

Recommendations

We were unable to complete a 3-Dimensional diffusion program for two major reasons. We have yet to have the complex math required for this type of program and the lack of time to complete a project of this type. If time allowed we could have written a more complex program that made it 3-Dimensional. If we had done this project later in our education, we would have used a diffusion formula that we could understand and tried to make the program 3-Dimensional.

Acknowledgements

At this time we would like to thank our mentor, Shaun Cooper, for helping us with our code and also understanding the code and how the gas diffuses, our teacher Jean McCray for choosing us to be in the class organizing our team with the choosing of our project and helping us with her knowledge. Our principal Kathy Vigil for providing us with the funds needed to on all our trips and allowing us to do this and also providing transportation, Los Alamos National Lab for providing the mode and pi computers allowing us to run our programs and also for hosting the supercomputing challenge and for all the people involved with the challenge, and all the scientists who lectured at the kickoff, Linda Riley who gave us a tour at NMSU engineering department, all the engineers and others that gave us advice and most of all the entire Challenge team for allowing us to be the first middle school to compete in the Supercomputing Challenge.

References

Okon Koko Ekpo Miami-Dade Community College, North Campus. Diffusion. (date not available) [Internet]. Available at, http://www.megaone.com/chemistry/gases11.htm

Prof. J Dana Eckart . Diffusion. (0ct 1997) [Internet] Available at, www.cs.Radford.edu/~dana/ca/examples/diffuse/diffuse.html

Prof. J Dana Eckart . (Feb. 2001) [E-mail interview] available at, *Dana@runet.edu* (Personal interview and tour of engineering departments)

Professor Linda Riley. (Feb. 2001) [Personal interview and a tour of the NMSU engineering facilities. Available at, Linriley@nmsu.edu

Methanol and methanol blends and their hazards) [Internet] www.Fta.dot.gov/library/technology/AFRISKS.htm

"Diffusion," Microsoft® Encarta® Online Encyclopedia 2000 http://encarta.msn.com © 1997-2000 Microsoft Corporation. All rights reserved.

Matheson Tri-Gas (2001) All rights reserved. http://www.mathesontrigas.com/msds//methane.htm#anchor-SYM41360

Appendix I

Appendix II

Phase I code

code

```
main()

{
  float M[10][10];
  float Down[10][10];
  float Across[10][10];

  int R, C;
  float mover, moved;

  int times;

  for(R=1;R<=5;R++)
   for(C=1;C<=5;C++)
     { M[R][C] = 0;
       Across[R][C]=0.4;
       Down[R][C]=0.4;
     }

  for (times=1;times<=5;times ++)
   {

       M[1][1]= M[1][1] + 100;

  for(R=1;R<=5;R++)
   {
   for(C=1;C<=3;C++)
     {
      mover=M[R][C] * Across[R][C];
      moved=M[R][C] * Down[R][C];
      M[R][C+1] = M[R][C+1] + mover;
      M[R+1][C] = M[R+1][C] + moved;
      M[R][C] = M[R][C] - mover -  moved;
     }
```

```
 }

for(R=1;R<=5;R++)
 {
 for(C=1;C<=3;C++)
   { printf(" %f ", M[R][C]);
   }
   printf("\n");
 }

   printf("\n");
   printf("\n");

}

                                                    }
```

Apendix III

Appendix IV

Phase II code

```
#include <stdlib.h>
#include <time.h>

main()


{
  float M[10][10];
  float Down[10][10];
  float RAcross[10][10];
  float UP[10][10];
  float LAcross[10][10];

  int R, C;
  float mover, moved,movel,moveu;

  float DISP, VA
  time_t now;
  int times;

  time(&now);


  srand(now);

 /* create the movement vectors */
  for(R=1;R<=5;R++)
   for(C=1;C<=5;C++)
     {    DISP=1.0; /*we can only disperse 100 % */

       M[R][C] = 0;  /* number of gas elements is 0 -- initial condition */


       VAL=rand() * DISP/RAND_MAX;
                          RAcross[R][C]=VAL;
```

```
    DISP=DISP -VAL;
    VAL=rand() * DISP/RAND_MAX;

    UP[R][C]=VAL;

    DISP=DISP -VAL;
    VAL=rand() * DISP/RAND_MAX;

    LAcross[R][C]=VAL;

    DISP=DISP -VAL;
    VAL=rand() * DISP/RAND_MAX;

    Down[R][C]=VAL;          }

/* Make the wall movements 0 */

for (R=1;R<=10;R++)
  { LAcross[R][1]=0;
    RAcross[R][5]=0;
  }

for (C=1;C<=10;C++)
  { UP[1][C]=0;
    Down[5][C]=0;
  }


/* run the system a set of time intervals */   for (times=1;times<=5;times ++)
 {

   /* inject the elements in the first location */
    M[1][1]= M[1][1] + 100;


  /* calculate the dispersion per unit time,
    unit time is defined as moving elements left to right top to bottom */
```

```
 for(R=1;R<=5;R++)
  {
  for(C=1;C<=5;C++)
    {
     /* calculate the amount of gas to move */
     mover=M[R][C] * RAcross[R][C];
                    moved=M[R][C] * Down[R][C];
     movel=M[R][C] * LAcross[R][C];
     moveu=M[R][C] * UP[R][C];

     /* move the elements */
     if (mover != 0.0 ) M[R][C+1] = M[R][C+1] + mover;
     if (moved != 0.0 ) M[R+1][C] = M[R+1][C] + moved;
     if (movel != 0.0 ) M[R][C+1] = M[R][C-1] + movel; if (moveu != 0.0 ) M[R+1][C] = M[R-1][C] + moveu;

     /* update the current cell */
     M[R][C] = M[R][C] - mover -  moved - moveu - movel;
    }

  }

 for(R=1;R<=5;R++)
                          {
  for(C=1;C<=5;C++)
    { printf(" %f ", M[R][C]);
    }
    printf("\n");
  }

    printf("\n");
 }

    printf("\n");
    printf("\n");

  }
}
```