Bowling Physical Modeling Program

Computer Systems Research Laboratory

2004-2005

Kevin Silas

June 20, 2005

Abstract

This paper reports on a project that studied the system of a moving bowling ball and its environment and produced a computer modeling program, as well as producing a series of conclusions about the system that came to light during and after the process. It was produced in C++, by first studying physical equations, applying them to the system, translating these methods into an iterative code, and then testing and implementing the resultant product. It combines the science of physics with the usefulness of computer modeling. The model does not attempt to predict with one hundred percent accuracy what will happen. It does, however, allow investigation into how changes in variables affect the final outcome. The Results section discusses these, along with future endeavors that may occur with this project.

1 Introduction and Background

Bowling is a sport that many people play at least once in their lifetime, but few take the time to practice and improve at it. It is actually quite complex, as are most sports. It takes an especially determined and skilled person to be able to bowl a perfect 300 game. Most bowlers learn early on that there are several manipulable variables that can help or hurt their game. Body placement, throwing angle, giving the ball some sidespin (known as hook in bowling terminology), and varying velocities can all improve ones game. But the question is, how exactly do these variable changes affect ones game, and by how much. Does sidespin impart more energy to the ball to knock the pins down? Does it change the spin of the pins upon impact, enabling them to knock over even more pins? Actually, spinning allows the ball to come into the pins at a greater angle, gives the bowler more control of the balls path, and altogether increases the range in which a ball can be thrown and impact successfully. Higher velocity provides greater power, but reduces a human bowlers ability to control the ball. The right angle is necessary to hit the sweet spot on the pins, and body position is an easy way to adjust the final position of the ball. There are many keys to the game of bowling, making it difficult to master.

Computer modeling of real-world systems is almost commonplace around the world today. Meteorologists use it to bring us weather forecasts, while Wall Street businessmen use it to try and maximize their returns. The problem with computer modeling is accuracy. As humans, we can derive all of the correct formulas and connections between variables, but the initial values can never be known precisely, or even to the preciseness necessary for our modeling formulas. Furthermore, some minor variables are often neglected, when even their small effects play a part over long time periods. For these reasons, the best use of computer models is not to predict the future, but to determine the sensitivities to variables in place in a system. It does not usually matter what the precise value of a variable is when determining the effect of changing it by some percent. That is why we cannot place our faith in the accuracy of models. Inherently, the modeling methods themselves also lead to errors. Generally, models have feedback loops ranging from the simple, time incrementation that changes the variables in step loop to the more complex, each variable independently changes, which affects the changes in one or more other variables loop. These loops only *approximate* continuous change, and often round off variables to too few decimal points.

In this bowling model, the situation being described was reduced to something quite simple out of necessity, and the model therefore neglects several small but significant variables. First of all, air resistance isnt computed, even though it can play a part in slowing the ball down, especially after it has stopped decelerating due to friction. Secondly, loft is assumed to be 0 under this model. Loft is the height above the lane that the bowler releases the ball. A higher loft means less hook, a lower speed, and possibly a change in direction upon contact with the surface. It is okay to neglect it here because professional bowlers often have a really small loft, which does not affect the path of the ball that much. These two neglected variables, along with several other minor ones and assumptions, add up to a small, but significant amount in the discrepancy between this model and a real-life situation. But the effect of changing the major variables is still essentially the same. The model may be able to come close to predicting the result of a ball roll, but it is more effectively used to investigate the systems sensitivity to variable changes.

There are two different levels of analyzing the physics of bowling. There is a simple way to calculate the forces involved if the bowl is going straight along a constant surface. The force that the ball experiences is simply the force of friction:

1.1 Friction Equation

F = -(coefficient of friction)*(mass)*(acceleration of gravity)

This force produces a torque on the ball about its contact point with the surface of the lane. This torque is simply the force times the radius of the ball. This torque divided by the moment of inertia of the ball ((2/5)*mass*radius*radius) is the angular acceleration. Once angular acceleration equals acceleration divided by radius, where acceleration is the force of friction (as above) divided by the balls mass, the ball is then rolling without slipping, with no spin in any direction but the direction of motion, and a magnitude to match the velocity of the ball divided by its radius. Although it looks somewhat complicated, this set of problem-solving steps is much simpler to solve for than that of a more typical bowling situation. In this case, the force of friction simply slows the ball down while at the same time spinning it forward, until it reaches an equilibrium position.

The problem becomes quite complex when two important qualities of the game are added in. Differing values of friction along the length of the lane are hazards for professionals and amateurs alike. The pattern of oil is rarely uniform or consistent. In this program there are several patterns to choose from to emulate this variable quantity in the real-world system. When this quality is added, the equation for rolling without slipping can no longer be solved using just formulas and algebra, because the coefficient of friction is changing, and therefore so is the force of friction. This makes it impossible to solve the problem as described above, and a modeling technique is required to figure out what is happening. Secondly, most bowlers above the amateur level apply at least a low angular velocity, or hook to their ball before sending it down the lane. This makes the physics problem even more complex, because there is an initial rotation that must be accounted for from the beginning, and the ball is now moving in two dimensions, which increases the complexity of the equations used. The straight-line problem involved multiplication of scalars (plain values), while the hook problem requires the use of vectors to describe each force, velocity, rotation, and position. In addition, in order to multiply these vectors, cross products must be used, for instance in the formula for torque. Setting up and visualizing the problem then becomes a challenge because of the different directions involved. When the ball doesnt move straight, it becomes much more appealing to look for solutions not found with paper and pencil. For both of these reasons, it is extremely convenient to use a modeling method to find solutions.

2 Development (Research, Work, Analysis)

2.1 Planning and Initial Research

The first step of this project was to discover what already existed in the world of computer bowling modeling and simulation. There are many simple, cheap games out there that look fun, but do not emulate the actual system in a good number of circumstances, such as certain split opportunities, the speed of the ball, angles, etc. A lot of these games calculate the results based not on physics, but on what experience tells the manufacturers usually happens. Some programs, such as Fast Lanes Bowling (see bibliography), do appear to have a grasp on the physical modeling side of the game. The program is based around a physics engine, giving it more credibility than the experience based games. There are also a number of similar bowling programs available that can also process the physics of the game fairly well. I knew that accurate physics was one of the main goals of this project, but I was not sure how far the model should extend.

When this program first began, it was substantially more ambitious. There was a plan for an accurate physics model of the entire bowling system, including all of the pins. Secondly the program would have an artificial intelligence portion in which an agent would choose the variables, review a database of results, and over time, learn a combination of variables that would result in maximizing strike potential. This proved to be too much, so the focus changed to just an environment, wherein the user would choose the values for the variables, and the output would be the resultant state of the pins. The physics of the pins also proved to be difficult (because of their unusual shape and moment of inertia), so the objective became to model only the ball as it traveled from the foul line to the position where the one pin would be. This was an interesting journey for this research project. The first step, after initial research, was to define exactly which problem I was going to conquer. As the journey progressed past the first stages of work on the project, the aim of modeling for prediction changed to modeling to find variable sensitivities, which turned out to be much more appropriate to the ideals of modeling. The entire process in fact has been dynamic, as there have been changes to the overall plan throughout the duration of research and development.

Before writing a modeling program, I had to research into modeling as a concept. As I have stated before, the purpose of modeling is analyzing variable sensitivities. I was introduced to this concept in my geosystems course, under the direction of Mr. Jarvis. After modeling the change in Earths temperatures over time with a program called Stella, it was made clear to the class that humanity is poor at predicting what will happen even in the near future. This point makes it easier to forgo several of the smaller parts of the system (or group them together) and only concentrate on several of the variables changing. For this program, the user can effectively change 6 variables from trial to trial. The first is the amount of friction that the ball encounters, and its placement on the lane. This is important to study because it varies widely from lane to lane and from bowling center to center. Professionals place a lot of emphasis on the distribution of lane oil because it is the one inconsistent variable that they cannot control themselves. The second is speed. Obviously bowlers can control this to some extent, so it is good to see the general effect of different speeds in conjunction with hook. The third is position, which a bowler can definitely control and obviously makes a difference on where the ball ends up in the end. Fourthly, bowlers can control their throwing angle, which in effect rotates the coordinates around a certain point. This makes for large changes in the resultant ball state, so it is important to consider. The fifth and sixth variables are magnitude and angle of hook. The magnitude is definitely necessary because at even low values, it can cause a huge change in the balls state. The angle is less important at low values, but can be important when emulating certain bowlers and/or bowling styles. These variables are sufficient to analyze almost all of the normal and purposeful changes that occur on a bowling lane.

2.2 Development and Codewriting

To actually create the modeling program, the physics of the system had to be thoroughly researched. I gained most of the information from being a student in Dr. Dells AP Physics class. Through that, and by studying the textbook, I complied a list of relevant equations from kinematics, dynamics, and rotational motion, and pieced together the relationships that each of them had with each other. This required some days of just studying the textbook, visualizing the changes of each vector variable, and I even used a bowling ball to help with this process. I also consulted with Dr. Dell early on, to make sure that nothing was missing and to learn his advice on how to go about writing the program.

Through my own decisions and Dr. Dells advice, I made a few assumptions concerning the model that made it many times simpler to work with, did not affect the outcome too much (which doesnt matter, as models are for studying variables, not predicting), and created some precise boundaries and limitations for the program to function in. First, the ball starts at the foul line every time, and always ends at the one pin line, exactly 60 feet down the lane. Secondly, as mentioned before, there are no loft nor air drag variables or equations present in the model. Thirdly, it is assumed that there are no deformities in the ball; it is perfectly round, with its mass distributed evenly, so that its center of mass is in the geometric center of the ball. Fourthly, the lane is assumed to be perfectly flat without deformities, and as soon as the center of the ball touches the gutter, it falls in. After making each of these assumptions, the program could easily be built without writing in any special cases, or worrying about the finger holes, or the chaotic behavior associated with releasing the ball at a height above the lane.

Before getting to the main part of the program, the physics engine, code had to be written that constituted functions that performed user interface, mathematical, menu, and database functions. Data was stored in a file by the trial in terms of roll number, and including information on the oil pattern used, initial and final speeds, initial and final angles, hook magnitude and angle, initial and final positions, and total elapsed time. The program used .10915 meters for the balls radius, 7.27 kilograms for its mass, and 9.81 meters per second squared as the constant acceleration of gravity. The main program starts with putting the data from the file into an array of structs, and ends with the rewriting of the file from the newly changed array. The menu gives 4 options: view database, perform a new trial, change the oil pattern randomly, or change the pattern to a specific style. Two math functions, one for magnitude of a vector and the other for angle, were used to help out, while a print function was used to print some of the data neatly. The database function simply printed out each trial in columns, while the change oil pattern function used a simple menu. Altogether, these steps were simple, with the exception of getting columns of data to line up neatly.

The other necessary part of the code besides those functions listed above and the physics engine was the oil pattern function. Each of the five patterns needed an equation in two variables to describe the relative strength of the coefficient of friction at a given point. A website was discovered that contained five different general patterns that commonly occur in a bowling alley. The diagrams can be found at the oil pattern website, given in the bibliography. The first one was split into two, both a short flat and a long flat version, which were simple to generate formulas for. The second and third ones (reverse block and crown) were a challenge, as I had to create a sine wave pattern that varied over the length of the lane. The fourth pattern, walling, was almost as simple as the first one, as it had sharp boundaries between values as well. The fifth pattern was dropped as it looked very similar to crown, and would not provide a difference in the variable sensitivity measurements. In addition to these diagrams of relative strength, there had to be a measurement or research into the absolute values of friction. The Columbia bowling website (also listed in the bibliography) provided some data to give an idea of the approximate friction values found on certain bowling lanes. From all of this, I was able to come up with a reasonable approximation to the average friction forces encountered on a bowling lane.

This led into the next, most difficult, and biggest step, which was to figure out how to translate the physics equations into an iterative format. This format would recalculate each variables value every thousandth of a second and use that new value for the next step. This entire process proved to be quite taxing, as there were numerous false starts, and positive/negative signs being misplaced are frustrating to deal with. Eventually however, a working, step-by-step process for getting new vector velocity and vector position values was found. After defining vector variables based on the user-inputted scalars, the program starts a time loop that ends when the ball reaches 60 feet. The program follows these steps: Gets the new coefficient of friction, gets the contact velocity from velocity, rotation, and the balls radius, calculates the force of friction which is in a direction opposite that of the contact velocity, finds if the contact velocity is sufficiently small to declare the ball rolling without slipping (which also causes friction to be 0), finds the torque, acceleration, and angular acceleration on the ball, adjusts angular velocity and velocity by their respective accelerations and the time increment, moves the ball according to the average velocity over the interval, saves the velocity magnitude and angle, checks to see if the ball is out of bounds, outputs the values if necessary and then updates the velocity. This process correctly moves the ball down the lane, one time-step at a time.

2.3 Errors, Testing, and Experimentation

While working on this program, I encountered several problems before coming upon the correct solution. I alluded to some of these earlier, but they bear mentioning again. It took a couple of weeks or so collectively to get all of the output functions working well. Getting the data to round off for output and line up correctly no matter what the value was somewhat of a hassle. Also, it took a while to realize that outputting during a trial needed to be regulated to get the right amount of outputs in case the speed was too high or too low. Outputting every .5 seconds is hazardous when the trial takes .1 second or 30 seconds. There was also, as I mentioned above, general trouble in deciphering the correct combination of equations and sequence of steps necessary to get the ball to move correctly. The problem that plagued me the most, however, was a matter of forgetting to change the velocity during the iterative loop. For several weeks, I could not figure out what was wrong, and why the balls variables didnt seem to change any of their values at all. Although this was a case of just plain idiocy on my part, it is important to mention all of the difficulties encountered.

Once the program appeared to be working correctly, it was necessary to test for accuracy. Initially, I just played with it for a while, and changed one or two values along the way to produce more realistic results. These changes did not interfere directly with the physics engine, but may have compensated slightly for the lack of other variables.

This was also the point where it was determined that hook needed to be split into two separate variables: magnitude and angle. This change allowed for more flexibility in the program. After all of these changes, I needed to formally test the program. A version of my program was developed that would test each variable with each other variable at 3 different values each. This matrix of 3 to the fifth, or 243 trials (which is shown in the appendix) was then examined to verify that no sets of values produced results contrary to human intuition of what should happen on a bowling lane. This visual analysis confirmed that the model was reasonably accurate and that the program was then finished.

The final planned stage of the program was to run an experiment testing the sensitivities of each of the variables when all else was constant, and try this on each oil pattern to see which variables changes the most. This plan was to involve 125 trials, with each variable being tried at five different values while the rest remained at a central value. Although a program was created similar to the testing version described above, but it was fundamentally flawed and I was unable to debug it in an adequate amount of time. Despite this unfortunate setback of gaining valuable information from my program, a few hours of casually playing around with the variables, only changing one from run to run for example, yielded some interesting results that will be further discussed in the next section. Some small variable changes could sometimes produce large swings in the resultant state of the ball.

3 Results and Conclusions

This project was a success! By the end of the year, a reasonably accurate model of the physical bowling ball system was produced. Currently, its usefulness is restricted to mostly variable sensitivities. Through casual observation during trial runs of the program, I have noticed several interesting things about variable behavior. I designed a standard run to test differences in variables: 18 miles per hour from the 25th board, at an angle of 1.5 and hook intensity .4 (no angle), on the short flat oil pattern. With the oil patterns, reverse block produces excess hook and is difficult to play, as expected. It caused a change of 10 boards, as opposed to the standard which produced just 2. I found the best oil pattern (3 boards-close to the one pin) on a standard run to be the walling pattern, which lives up to its title as adult bumper bowling. In terms of speed, a change of just 1 mph (with a everything else standard) produced a change of 5 boards. Position is the easiest variable to predict changes with. A board to the left change produced approximately a board to the left change in most instances. In terms of spin, a change of just .1 of hook magnitude caused about a 10-board change. This was found to be the most sensitive variable because of the wide changes it causes. This is not unreasonable, considering the much higher levels of success hook bowlers have compared with straight bowlers. Bowlers realize that learning to control this sensitive variable will allow them to hit any range of positions, and do so at higher speeds and sharper angles, leading to increased strike percentages. Ultimately, this journey has taught me that to get better at bowling, or any similar sport or game, one must concentrate on controlling and realizing the sensitivities of relevant variables. There are some that cannot be changed, while others vary without any apparent effort.

This project path has potential for further study. The previously neglected variables can be added, the time increment can be changed so that the model becomes more accurate. More complex oil patterns can be designed. Pins can be added in the future to study the impacts and the complexity of the interactions. Ultimately, it may be possible to achieve my initial goal of having an artificial intelligence component that can master the game. There is already a robot capable of imitating most professionals bowling styles, which is used for testing new bowling balls. Perhaps it will one day figure out how to bowl a 300 game.

4 Acknowledgements of Assistance (Bibliography)

I would like to thank Mr. Jarvis and Dr. Dell for their pieces of advice relating to my project, about the purpose of models and the physics of bowling, respectively. I would like to thank Mr. Latimer and Mr. Strong, for helping me realize my project goals. For research, I used Tipler's *Physics for Science and Engineers* as a physics guide, as well as the follwing websites:

1. Fast Lane Bowling: http://www.labratsgames.com/labrats.html
2. Lane Diagrams here:
   http://ourworld.compuserve.com/homepages/kennmelvin/tOil.htm
3. Friction Values can be found here:
   http://www.columbia300.com/innovation/techdocs.cfm?id=20
4. Interesting Diagrams: http://www.mrcla.com/bowling/bowling-pres/index.htm