Math and Science

Help Topics

Introduction

In **Math and Science**, the first five programs are aimed at Grades 1 through 6 and help teach the four basic arithmetic operations of addition, subtraction, multiplication, and division. The last five programs present real-world applications of physics and math - great ideas for science fairs! For kids aged 9 to 16.

The first four programs give you the option of solving a randomly generated problem or one you enter. You also have the option of solving the problem yourself or having the computer show you, in a step by step fashion, how to solve the problem. In **Addition Tutor**, you get practice in adding from 2 to 6 numbers (with 1 to 6 digits each). **Subtraction Tutor** provides column subtraction problems. The program gives optional practice in 'borrowing' concepts. **Multiplication Tutor** helps you learn how to multiply numbers with up to five digits. You enter your answers just like you work them on paper - column by column. In **Division Tutor**, you obtain help in solving long division problems. Problems are solved just like you would on paper and remainders are optional. In **Flash Card Math**, you get practice in all the above skills. The level of problem difficulty can be varied and, based on performance, the computer will encourage moving to different skill levels. Also has optional timer.

Circuits allows you to build electric circuits from batteries, switches, and light bulbs. See effects of different circuit types, burned out bulbs, etc. **Levers** teaches you about the three different lever types and forces required to balance them. **Moon Lander** gives practice in understanding concepts of speed and acceleration as you try to land on the moon. Study basic oscillations with **Pendulum** -- see effects of changing lengths, weight, and friction. Plots of motion are given. **Water Wheel** demonstrates the exciting world of chaotic dynamics. Examine wheel motion as different water flow rates are used.

Addition Tutor

In this program, your child learns the basics of addition. The program is run by selecting **Addition Tutor** from the main menu screen. You have three choices: **Solve Problem**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Solve Problem:

Prior to solving an addition problem, you set up the problem by selecting the **Problem Mode**. Decide between **You Input Problem** or **Computer Selects Problem**. If you choose **You Input Problem**, you will be asked to enter from 2 to 6 "addends" (the numbers to be added together). Each addend can have a value from 0 to 999999. To do this, type the addends in any of the six spaces provided. If you choose **Computer Selects Problem**, you must make two other choices. Choose **How Many Numbers?** you want to add together - choose from **2** to **6**. Next, choose the largest **Number of Digits** you want in the numbers to be added - select from **1** to **6**. Once a problem is entered or the computer selects one, you can begin solving your addition problem by clicking **Solve Problem**.

The addition problem to be solved will be displayed. (If you input the problem, the addends may have been reordered.) Also displayed are four control buttons: **Solution Help**, **Program Help**, **Stop**, and **Carry**. You solve the problem just like you would on paper. That is, add the given numbers column by column, starting from the right and going to the left. A question mark (**?**) is used to mark the column you are working on. You select numbers by typing them or by clicking on the displayed keypad. If your selection is correct, it will be displayed. If it is wrong, an uh-oh sound will be heard and you must try again. If the sum in any column is larger than 10, you must 'carry' to the next column before continuing. To do this carrying, simply click the **Carry** button or press the **C** key and then the number to carry to the next column. At any time in the solution process, you may click the **Solution Help** button. Doing this tells the computer to show you, in a step-by-step manner, how to solve the remainder of the problem. At any time in the solution process, you can click the **Stop** button to return to the **Addition Tutor** menu screen.

Once the problem is finished, you click **Next Problem** to continue. If you input the solved problem, you will be returned to the **Addition Tutor** menu screen. If the computer chose the problem, another problem will be displayed for solution.

Help:

Clicking **Help**, **Program Help** (while solving a problem), or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Subtraction Tutor

In this program, your child learns how to subtract (including borrowing, if desired). The program is run by selecting **Subtraction Tutor** from the main menu screen. You have three choices: **Solve Problem**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Solve Problem:

Prior to solving a subtraction problem, you set up the problem by selecting the **Problem Mode**. Decide between **You Input Problem** or **Computer Selects Problem**. If you choose **You Input Problem**, you will be asked to enter a minuend (the top number) and a subtrahend (the bottom number). Each number can have a value from 0 to 999999. To do this, type the numbers in the spaces provided. If you choose **Computer Selects Problem**, you must make two other choices. Choose the largest **Number of Digits** you want in the numbers to be subtracted - select from **1** to **6**. Also, choose whether you want **Borrowing** in your problem. Once a problem is entered or the computer selects one, you can begin solving your subtraction problem by clicking **Solve Problem**.

The subtraction problem to be solved will be displayed. Also displayed are four control buttons: **Solution Help**, **Program Help**, **Stop**, and **Borrow** (if borrowing selected). You solve the problem, column by column, starting from the right and going to the left. A question mark (**?**) is used to mark the column you are working on. You select numbers by typing them or by clicking on the displayed keypad. If your selection is correct, it will be displayed. If it is wrong, an uh-oh sound will be heard and you must try again. If the lower digit is larger than the upper digit, you must 'borrow' 10 from the preceding column before continuing. To do this borrowing, click the **Borrow** button (or press the **B** key) and then enter the new minuend (top number) values as they are requested. For example, if the upper digit in one column is 4 and the lower digit is 6, you first borrow. You decrease the next column's minuend value by 1 (depending on number values, you may also be asked to decrease column values in other columns - just remember to watch for any requested values). You then type a 1 and a 4 to indicate the upper number is 14 after the borrowing. At any time in the solution process, you may click the **Solution Help** button for assistance. Asking for such assistance tells the computer to show you, in a step-by-step manner, how to solve the remainder of the problem. To stop solving problems, click the **Stop** button. This will return you to the **Subtraction Tutor** menu screen.

Once the problem is finished, click the **Next Problem** button to continue. If you input the solved problem, you will be returned to the **Subtraction Tutor** menu screen. If the computer chose the problem, another problem will be displayed for solution.

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Multiplication Tutor

In this program, your child learns how to multiply two numbers together. The program is run by selecting **Multiplication Tutor** from the main menu screen. You have three choices: **Solve Problem**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Solve Problem:

Prior to solving a multiplication problem, you set up the problem by selecting the **Problem Mode**. Decide between **You Input Problem** or **Computer Selects Problem**. If you choose **You Input Problem**, you will be asked to enter a multiplicand (the top number) and a multiplier (the bottom number). Each number can have a value from 0 to 99999. To do this, type the numbers in the spaces provided. If you choose **Computer Selects Problem**, you must make one other choice. Choose the largest **Number of Digits** you want in the numbers to be multiplied - select from **1** to **5**. Once a problem is entered or the computer selects one, you can begin solving your multiplication problem by clicking **Solve Problem**.

The multiplication problem to be solved will be displayed. Also displayed are four control buttons: **Solution Help**, **Program Help**, **Stop**, and **Carry**. Multiply the given numbers, column by column, forming each required subproduct, then sum the subproducts to get the final product. A question mark (**?**) is used to mark the number position you are working on. You select numbers by typing them or by clicking on the displayed keypad. If your selection is correct, it will be printed. If it is wrong, an uh-oh will be heard and you must try again. If the product of any two digits or the sum of digits in any subproduct is larger than 10, you must 'carry' the proper number to the next column before continuing. To do this carrying, click the **Carry** button (or press the **C** key). When multiplying, you then type in the number to carry to the next column. When adding subproducts, the computer fills in the carry value for you (saving you a little time - the computer assumes youve studied the **Addition Tutor** program enough to know adding by now). At any time in the solution process, you may click the **Solution Help** button for help. Asking for help tells the computer to show you, in a step-by-step manner, how to solve the remainder of the problem. To stop solving problems, click the **Stop** button. This will return you to the **Multiplication Tutor** menu screen.

Once the problem is finished, you click the **Next Problem** button to continue. If you input the solved problem, you will be returned to the **Multiplication Tutor** menu screen. If the computer chose the problem, another problem will be displayed for solution.

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Division Tutor

In this program, your child learns how to do long division. The program is run by selecting **Division Tutor** from the main menu screen. You have three choices: **Solve Problem**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Solve Problem:

Prior to solving a division problem, you set up the problem by selecting the **Problem Mode**. Decide between **You Input Problem** or **Computer Selects Problem**. If you choose **You Input Problem**, you will be asked to enter a dividend (the number being divided) and a divisor (the number being divided into the dividend). Each number can have a value from 1 to 999999. To do this, type the numbers in the spaces provided. If you choose **Computer Selects Problem**, you must make two other choices. Choose the largest **Number of Digits** you want in the divisor and quotient (the answer) - select from **1** to **4**. Then, select whether you want **Remainders**. Once a problem is entered or the computer selects one, you can begin solving your division problem by clicking **Solve Problem**.

The division problem to be solved will be displayed. Also displayed are several control buttons: **Solution Help**, **Program Help**, **Stop, Carry, Borrow, Digit Down**, and **Remainder**. Divide the given numbers using the five steps of division: 1-Divide, 2-Multiply, 3-Subtract, 4-Digit Down, and 5-Do it again. A question mark (**?**) is used to mark the number position you are working on. You select numbers by typing them or by clicking on the displayed keypad. If your selection is correct, it will be printed. If it is wrong, an uh-oh sound will be heard and you must try again. When first starting to solve a problem, you must tell the computer which digit in the quotient you are solving for. The possible starting columns are marked by boxes. Click the desired starting position. When multiplying numbers, if the product of any two digits is larger than 10, you must carry the proper number to the next column before continuing. To do this, click the **Carry** button (or press **C**). The computer will fill in the carry value (we assume youve had enough practice in the **Multiplication Tutor** program). Likewise, when subtracting if the lower digit is larger than the upper digit, you must borrow 10 from the preceding column before continuing. This is done by clicking the **Borrow** button (or pressing **B**). To speed up the solution process, new column values will be computed and displayed for you; that is, you don't have to enter them like you do in the **Subtraction Tutor** program. To bring a digit down from the dividend, after the subtraction step, just click the **Digit Down** button (or press **D**). If there is a remainder at the end of the solution, click the **Remainder** button (or press **R**), then enter the required numbers. At any time in the solution process, you may click the **Solution Help** button for help. Asking for help tells the computer to show you, in a step-by-step manner, how to solve the rest of the problem. To stop solving problems, click the **Stop** button. This will return you to the **Division Tutor** menu screen.

Once the problem is finished, you click the **Next Problem** button to continue. If you input the solved problem, you will be returned to the **Division Tutor** menu screen. If the computer chose the problem, another problem will be displayed for solution.

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Flash Card Math

This program provides practice in answering basic addition, subtraction, multiplication, and division problems. The program is run by selecting **Flash Card Math** from the main menu screen. You have three choices: **Solve Problems**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Solve Problems:

Clicking this button will generate random problems, based on selected options. **Problem Type** is selected by clicking on either **Addition**, **Subtraction**, **Multiplication**, or **Division**. Problem **Difficulty Level** (Levels **1** through **6**, 6 being the hardest) is also selected, as is your choice under **Timer On?**. Once all choices are desired, you can begin solving problems.

You are given a problem. Enter your answer (the computer shows you how many digits are in the answer). You select numbers by typing them or by clicking on the displayed keypad. If correct, a tune is heard and, after a short delay, the next problem is generated. If incorrect, an 'uh-oh' is heard. If the timer is off, you have three tries to get the correct answer - after your final try, the correct answer is given. If the timer is on, you can answer until time (10 seconds) runs out. The fewer tries you need to answer a problem, the higher your score. The program will generate problems until you click the **Stop** button. Once you stop, your final score is given along with an evaluation of your performance (suggestions for moving to different levels are given if you solved at least five problems). Click **OK** to return to the **Flash Card Math** main screen.

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

About Simulation

The next five programs in the **Math** and **Science** series (**Circuits**,**Levers**, **Moon Lander**, **Pendulum**, and **Water Wheel**) demonstrate a principle used extensively in engineering and mathematics disciplines, that of computer simulation. Simulation is used to test designs and procedures before actually building some device or structure. For example, we use simulators to train airline pilots - this is much safer than training them in actual planes because if they make a mistake, there is no loss of airplane or life. Before constructing a building, we simulate it on a computer to see if it can withstand winds and possible earthquakes. Using simulations, we can determine how a system operates, learn how its performance varies as we change different parameters, and develop operation procedures.

Five simulations are covered: basic electrical circuits, three types of levers, a lunar landing vehicle, a simple pendulum, and a special type of water wheel. We will look at how these simulations work (including how to use the corresponding programs) and how they could be used within a classroom or home learning situation. In each simulation, we have attempted to use realistic numbers which would allow the more industrious of you to build the actual system and see how it compares to the computer model.

Circuits

In this program, you can build electrical circuits using batteries, switches, and lights. By flipping switches and burning out light bulbs, you can see how the circuit you build works. The program is run by selecting **Circuits** from the main menu screen. You have three choices: **Edit Circuit**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Edit Circuit:

The program begins in this mode - the last saved circuit configuration is displayed. You can click **Clear Circuit** to clear the current configuration. But, be careful, you cannot Undo a circuit clearing. There are nine possible locations for circuit elements, each indicated by a box. You pick a desired location by clicking with the mouse. If you click on an empty location, a pictorial menu appears giving you five element choices: **Battery**, **Inverted Battery**, **Light**, **Switch**, or **Wire**. Click on the desired choice. That element will be placed in the circuit. If you click an occupied space, the element there is erased before the selection menu appears. You are limited to two batteries, five switches, and five lights in your circuit. Once these limits are met, the corresponding menu option is disabled. We suggest playing around with the circuit editor the first time you use the program to get used to its operation. Click **Cancel** in the menu, if no choice is desired.

Once your circuit is as desired, click **Simulate Circuit** - the screen changes to the simulation screen. The program computes the voltages and currents in the circuit and the flowing paths are colored in red - paths without current flow remain yellow. If currents flow through OK lights, they light up, brightly for larger currents, dimly for small currents. Voltages at five different locations in the circuit are indicated (each battery is assumed to be a common 1.5 volt cell and the bottom of the circuit is assumed to be ground or zero volts). At this point, you may make changes in the light/switch status and see the results. To make a change, click on the appropriate circuit element. If you click an OK light, it changes to a burned out light, and vice versa. If you click a Closed switch, it opens, and vice versa. After each change, new currents are computed and the circuit status is updated. Please note that simulating the circuit involves some detailed math (solving lots of equations using matrix inversions), hence each time something is changed a certain amount of time is needed to do the calculation (a couple of seconds on a 386-based PC). To return to the edit screen, click the **Stop** button. Let's look at some ideas of how this program can be used.

Basic Concepts

An obvious use for the program is to study the basic concepts of current flow and voltages. A great analogy to use is water flow in pipes, where pressure corresponds to voltage and flow corresponds to current. Current (water) can only flow through complete paths. The higher the voltage, the larger the current. Read about resistance and how it affects voltage and current. Study short circuits - if a wire is connected in parallel with a light, why does the current flow through the wire and not the light? (Answer less resistance). Learn about Ohm's law that says voltage = current times resistance - show why short circuits are dangerous (low resistance, high voltage means high current).

Circuit Types

There are two basic types of circuits: series and parallel. First simulate a circuit with light bulbs in series. See what happens when a light burns out. Do the same for a parallel connection. Explain the results. Try simulating short circuits. What's the difference between two batteries connected in series and two in parallel?

Build Circuits

Get some actual batteries, lights (flashlight bulbs work great), switches (use paper clips screwed to a block of wood), and wires. Build some real circuits and play around with them. Study parallel and series circuits. Get a voltmeter (try an electronics store) and measure voltages at different points in your circuit. Try to get some concept of resistance - your voltmeter can be used to measure the resistance of different wires, lights, and other objects. See what items are good conductors (low resistance) and which are not (high resistance).

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Levers

With this program, you can study the three basic lever types and the balancing (lifting) forces required. The program is run by selecting **Levers** from the main menu screen. You have four choices: **Learn About Levers, Work With Levers**, **Options**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Learn About Levers:

With this option, you will be taken through several screens of information defining basic terminology (fulcrum, effort, load), the idea of mechanical advantage, what effort is required to balance a known load, and examples of the three types of levers (1st, 2nd, 3rd Class). You click on **OK** to move from screen to screen. Click the **Stop** button to stop the review.

Work With Levers:

With this option, the last saved lever configuration will be displayed and the current values for lever type, effort, load, and distances will be printed. You have several options at this point: **Change Load**, **Move Effort**, **Move Load**, **Move Fulcrum**,or **Apply Effort**. You can also click the **Stop** button and return to the main menu.

To change the load, move the effort, move the load, or move the fulcrum, simply use the corresponding scroll bar to make the desired change. The current values are updated as you make changes. Once the lever configuration is as you desire, click **Apply Effort**. Use the displayed scroll bar to increase or decrease the effort. The current value is updated as you make changes. A maximum value of 200 is allowed and a minimum of 0. When the effort is near the value required for balance, the lever will begin to move. If the effort equals the balance value, the lever will eventually level out. Note that sometimes (since the effort can only be changed in increments of 1) that the exact balance value may not be attainable. In this case, the lever may just oscillate about the level condition. Click **Change Values** to return to the load, effort, and fulcrum editing screen.

Here are some ideas of what to do with the lever simulation.

Basic Concepts

Discuss the ideas of forces and moments (force times distance). Explain how moments cause turning actions. Discuss the idea of mechanical advantage - and why sometimes a mechanical disadvantage is better. How does friction in a lever come into play? When is a lever without any effort or load balanced? Discuss how the weight of the lever itself could be added into the balance equation (as written in the program, the balance equation assumes a weightless lever). Discuss the advantages and disadvantages of the three different lever classes.

Identifying Levers

See how many levers of each of the three classes you or your child can identify in everyday surroundings. For each, describe what the load, effort, and fulcrum are. Does each have a mechanical advantage or disadvantage or no advantage?

Building Levers

With a board, something for a fulcrum, and a few weights, it is quite easy to build your own lever system. Build the different classes. Measure forces using something like a spring scale. Examine balance conditions and mechanical advantage.

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Moon Lander

You are the pilot of a lunar landing vehicle hovering over the moon. You must land it safely using your thrusters. The program is run by selecting **Moon Lander** from the main menu screen. You have three choices: **Start**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Acknowledgment: This program was adapted from one described in the book Black Art of Visual Basic Game Programming, by Mark Pruett, published by The Waite Group Press, 1995.

Start:

Prior to starting your descent to the moon, choose your pilot level: **Auto**-**Pilot**, **Beginner**, **Novice**, **Junior**, **Senior**, or **Advanced**. And, select the type of units: **US** (feet) or **Metric** (meters).

On the simulation screen are several important pieces of information. The vehicle is shown on the left side. The vehicle moves both vertically (up and down) and laterally (left and right). At the bottom of the vehicle display are two small boxes, one green and one red. The red box indicates the landing pad lateral location, while the green box indicates the moon landers current lateral location. The right side of the screen displays elapsed time (**Time**), altitude value (**Altitude**), descent speed, lateral position relative to the landing pad (**Position**), and lateral speed. Also shown are a plot of current vehicle position relative to the landing pad (with a suggested straight-line, descent trajectory), fuel level, and the thruster controls. The screen displays are updated every 0.5 seconds.

The idea under each option is the same. Your vehicle is some height above the surface of the moon at zero speed. Due to gravity (about one-sixth the gravity on earth), you begin to accelerate toward the surface with a corresponding change (goes negative) in your speed (a negative speed indicates descent - positive speed indicates ascent). The current altitude and speed are always displayed. You slow your descent by clicking on the retro thruster (the one with a down arrow). This thruster counteracts gravity effects. You must also adjust the landers lateral position relative to the landing pad. Your current lateral position (- if to the left of the pad, + if to the right) and lateral speed (- if moving to the left, + if moving to the right) are displayed. Lateral motion of the lander is controlled using the two thruster controls with left and right arrows. Note the left thruster moves the lander to the right and the right thruster moves the lander to the left. Since there is no atmosphere on the moon to create friction, the only way to slow lander lateral motion is to use the opposite thruster. All thrusters operate in impulse mode. That is they just fire momentarily, changing the lander speed in the corresponding direction. In rocket scientist talk, we say the thrusters impart a delta V (change in velocity) to the lander.

The idea of the program is to be over the landing pad and going at a slow descent speed (and slow lateral speed) by the time your altitude (height above the moon surface) reaches zero. A safe landing is one where the final speed is between 0 ft/s and -10 ft/s (-3.0 m/s); higher descent speeds result in a crash! There are no crash penalities for missing the pad - your astronauts will just have a long walk back to the home base!! To illustrate the use of these controls, choose the **Auto-Pilot** option and watch the computer land the vehicle - it does a fairly decent job. Click **Start** to start the simulation. At any time during the program, you may click **Pause** to pause the action (click **Resume** to restart) or click **Stop** to stop the current run.

In the **Beginner** option, you get practice controlling the vehicle in the lateral direction. The computer will control the retro thruster, you only use the left and right thrusters. Try to follow the suggested trajectory on the display and keep your lateral speed fairly low. Keep trying this option until you feel comfortable getting the lander on its pad. The **Novice** option provides practice in slowing the landers vertical speed. The lander is placed directly over the landing pad and you adjust vertical speed using just the retro (down) thruster -- you only have to worry about final speed. Practice this option until you rarely crash the lander.

The **Junior** option moves you up to having the ability to control all thrusters, and hence all motions on the lander. You now have to position the lander over its pad and make sure you are going at a slow vertical speed when landing, or else. Then, the **Senior** option offers one more concern. Up to now, we assume you have unlimited fuel for thruster use (that is, the fuel gauge never changes). With the **Senior** (and **Advanced**) option, the fuel gauge shows you how much fuel you have remaining. Each thruster burn uses some fuel (the retro thruster uses twice as much as the left and right thrusters). So now, you must land at a safe speed, over the landing pad without running out of fuel. At low fuel levels, a beep will be heard. Once you run out of fuel, your vehicle will just accelerate toward the lunar surface, most likely resulting in a crash. The final level is **Advanced**, where system malfunctions are simulated. Your thrusters work in a random manner and your fuel tank leaks. When the thrusters are used at this level, the amount of thrust is not at all predictable. Constant adjustments are needed to keep the vehicle over the pad and within speed limits.

Try each level and see what it takes to master landing your vehicle. After each landing, evaluate your performance based on elapsed time, final descent and lateral speeds, and how close you are to the pad. Always try to improve on the last run. This program offers a lot of potential for further study and experimentation.

Basic Concepts

Much can be taught about just the concepts of speed, position, acceleration, and gravity. Explain how the retro thruster works to counteract gravity and how the lateral thrusters move the vehicle. Talk about Newtons Law. Derive the basic equations that show the relations between altitude, position, speed, and acceleration.

Landing Techniques

It will become immediately obvious that there are many ways to obtain a safe landing with the vehicle. Without a fuel constraint, you can take forever to land. A more interesting problem is to try to land as fast as possible. This is the so-called minimum time problem. Or try to land using the least fuel possible - an important problem in space travel where you don't want to launch a lot of unnecessary weight.

Try a so-called coast and burn approach. That is, let your lander keep dropping until some point where you apply nearly constant retro thrust until landed. This approach needs to be used in space vehicles where thrusters can only be turned on, but never off (like solid rocket motors).

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Pendulum

This simulation is that of a simple pendulum. The pendulum is displaced some amount from vertical and released. Its corresponding motion can then be studied. Such a system is a good introduction to the concepts of oscillation and frequency. This pendulum model assumes that the mass of the blob at the end of the pendulum is much greater than the mass of the connecting link and that there is some adjustable amount of friction in the pendulum pivot. The program is run by selecting **Pendulum** from the main menu screen. You have three choices: **Start**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Start:

Current pendulum conditions (**Angle**, **Mass**, **Length**, **Friction**) are shown and displayed. Prior to starting the pendulum motion, you can change any condition you choose. To change a value, use the mouse to click on the corresponding arrow controls to increase or decrease the value. There are limits on what values can be used. The angle can vary from 0 to 45 degrees, mass can vary from 10 to 5000 grams (0.1 to 10 pounds), length from 50 to 200 cm (10 to 80 inches), and friction from 0.00 to 1.00 (0 being no friction and 1.00 a large amount of friction). You can also select the type of units used. Make your choice between **US (English)** units (pounds, inches) or **Metric** units (grams, centimeters). Note if you change unit type, current values (pendulum length and mass) will be changed to the new units. The changed values will not be exact equivalents of the previous values though due to some adjustments made by the program. Once the values are as desired, click on **Start** to start the simulation.

During the simulation, the animated pendulum motion is seen and in the upper right corner, a plot of the angle the pendulum makes with vertical (as time on the horizontal axis increases) is drawn. In the lower right corner is a 'phase-plane' plot of pendulum motion. On the horizontal axis is pendulum angle and on the vertical axis is the pendulum rotational speed. The simulation runs for 20 seconds or until **Stop** is clicked. At the end of the simulation, the total time is printed. You can rerun the simulation any number of times - click **Reset** to return to the value editing screen.

So, what can you learn using such a simulation? There are several possibilities, depending on grade level. Here are some ideas - use whatever you think might be useful.

Pendulum Operation

First, it is valuable just to have students describe how a pendulum works - what makes it oscillate, how does friction affect the motion? It's all related to potential and kinetic energy and losses associated with the friction. Will more friction slow it down or speed it up, why? If the friction could be increased to infinity, what would happen? How is motion affected by length? mass? initial angle? How are the plots (angle response, phase plane) affected by different parameters? Can students describe what's going on in the phase plane plot? Where is speed highest? Lowest? Relate these answers to physical reality. Can older students write down equations that help describe pendulum operation? Where are pendulums, or similar oscillating devices, used?

Oscillation Frequency

For a given configuration (angle, mass, length, friction), compute the frequency of oscillation. The frequency is equal to the number of cycles completed during the simulation divided by simulation time. The unit of measurement equivalent to cycles per second is a Hertz, abbreviated Hz.

Frequency Dependencies

For a given value of friction, how does the frequency vary with length and/or mass. Plot the results of your experimentation. What if you need a certain frequency - how could you find out what length/mass combination gives you that value? Use interpolation on your plot for such a computation. How does frequency vary as the friction is changed? Is frequency more dependent on length, mass, or friction?

Real Time vs. Computer Time

The time printed on the screen is simulation seconds, or what we call real-time. That is, it is the actual time it would take the pendulum to complete its motion if we were using the real system. It is not the time required by the computer to compute its solution. The time needed by the computer to compute its solution depends on how fast your computer is. Most newer computers (486 processors and later) are so fast that the program is purposely slowed down to make sure the computation time approximates real time. If you are using a slower machine, computation time will be greater than real-time. In simulations of dynamic systems, we must distinguish between computer time and real-time. This concept could be explained to students to initiate a discussion on simulation. In most cases, the computer is fast enough that computer time is less than real-time, i.e. the computer can compute its solution faster than the real system. In some cases, however, the simulation is so complicated that it might take a computer a couple of hours just to compute a few seconds of real-time results. And there are cases where we want computer time to equal real-time, for example with airplane simulations.

Building a Pendulum

It might be fun to build a real pendulum and see how its performance relates to the simulated pendulum. I would suggest just hanging a weight (mass value measured in grams) on a string (remember we assumed the weight of the blob is greater than the connecting link). Using a stop watch, determine how long it takes to complete 10 complete (back and forth) cycles and compute the frequency of your pendulum. Measure the length - it can't be longer than 200 centimeters (80 inches). Enter the measured mass and length in the simulation with Friction=0 and run the program. See how that compares to your measurement. Can you match the computed frequency with the computer by changing parameters? (The length you need in the computer model may differ from the actual length - this is because we assume the connecting link of the pendulum has no mass.) How can you estimate the amount of friction in your pendulum using the computer model? How does the oscillation time vary as you change the length? I'm sure you can think of more experiments of this type. Using a real system to check on a computer simulation is called model validation. It is probably the most important part of computer simulation - for example, it would be worthless to use airplane simulators to train pilots if they didn't accurately portray how an airplane works and responds.

Measuring Gravity (Advanced Topic)

If you build a pendulum (of length L), you can use it to measure the earth's (or any other planet you could travel to) gravitational acceleration. To do this, we first find the period of the pendulum - that's the time it takes to complete one oscillation cycle. The best way to find the period is to displace your pendulum some amount and let it go. Use a stopwatch to time how long it takes to go through 10 complete cycles. Divide that time by 10 and you have an average period value - call that value T (in seconds). Now you need the formula for gravity (G). We'll just give it to you:

G = 4(3.14159)^2L/T^2

This formula is read as 4 times Pi-squared times L divided by T squared. G will come out in units of cm per second per second (metric units) or in per second per second (US units). These are the units of acceleration (the rate of change of speed). As an example, say you have a pendulum that's 100 cm long. It takes 20 seconds to complete 10 cycles, so its period is T=2 seconds. Then G is

G = 4(3.14159)^2(100)/(2)^2 = 987 cm/sec per sec

Look up the actual acceleration in a reference and compare your value with the actual value. Discuss possible error sources. You must be very exact in your timing of the period. Also make sure the weight on the end of your pendulum is much heavier than the connecting link of the pendulum. A good suggestion is to hang a heavy weight on the end of a long string.

Inverted Pendulum

Another type of pendulum we look at in the engineering world is an inverted, or upside down pendulum. The mathematics involved in balancing an inverted pendulum are used in every rocket and missile guidance control system ever designed. Have students describe how a pendulum would work if it were turned upside down (of course, the connecting link for an inverted pendulum could not be a piece of string!). Make some inverted pendulums and try to balance them. Can the students describe in some mathematical sense what logic they use in balancing the pendulum - this is what simulation specialists have to do in order to program a computer to do rocket guidance. Are longer pendulums easier or harder to balance than shorter ones? Hint: have you ever seen a short, fat rocket?

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

Water Wheel

This final simulation is quite interesting. In most systems, we can predict how they will perform given the operating parameters. For example, we know if we displace the pendulum (in the **Pendulum** program) from vertical and release it, it will oscillate and eventually stop (assuming some friction). If we pull it back a little more, it will oscillate longer but still it will stop. Would you be surprised if we pulled the pendulum back some more and it didn't oscillate, but instead swung completely around its pivot and rotated wildly? I think so. Of course, we can't do this with the pendulum, but there are some dynamic systems where small parameter changes can result in very different, counterintuitive, overall behavior. One classification applied to such systems is that of "chaotic system." The water wheel examined here is a famous chaotic system known as Lorenz Water Wheel. The program is run by selecting **Water Wheel** from the main menu screen. You have four choices: **Start**, **Parameters**, **Help**, or **Exit**. Make your choice by clicking on the desired button.

Start:

The water wheel is drawn on the left side of the screen. Note the wheel consists of a number of sticks that rotate in a vertical plane about its center. On the end of each stick are cans that can be filled with water. The weight of these cans cause the rotation of the stick. A good mental picture of such a wheel would be a motorless ferris wheel with a seat on each end of a stick. The filling source is at the top of the wheel, so as the cans pass under the source they are filled with water. Note the longer a can stays under the source, the more water it is filled with it. The last piece of information needed is that each can has a hole in it, so the water is constantly draining and refills are needed to maintain some rotation. So, in principle, the wheel works this way: a can is filled which causes some rotation. As it rotates, water is lost necessitating refill. As long as water remains in either or both cans, there is some motion. The kind of motion imparted to the water wheel is where the chaos comes in.

You start the water wheel simulation by clicking **Start**. Then, during the simulation, you only control one parameter - the value of the input flow source. This is decreased and / or increased by using the horizontal arrow control near the top of the screen. The current flow is printed (grams per second for metric units, ounces per second for US units) near the flow source, along with a bar chart indicating what proportion of the maximum flow is being used. Several displays record the wheel motion. In the upper right corner is a display of rotational speed. Values above the line indicate a counterclockwise rotation, while values below the line show clockwise rotation. The display immediately under this curve shows the vertical height of one of the cans (the one initially at the top of the wheel). And the final curve shows the level of the water in each of the cans. These curves will show the chaos implicit in the water wheel system. Once the flow is set, the wheel will turn in one direction, change speed, rotate in another direction, and maybe even stop. Completely different responses occur with different flow rates. The simulation can be stopped and restarted by clicking the **Restart** button.

To halt the simulation and allow further parameter changes, click the **Stop** button.

With all this explanation out of the way, what do you do with the water wheel simulation? Our suggestion is play with it and be amazed and astounded at its weird behavior. Try all kinds of different physical parameters and flow rates. Can you ever get the wheel to spin at a constant rate for a long period of time, like a water wheel is supposed to work? Use the simulation to learn and study concepts like rotational inertia, friction, torques, and flows. Can the advanced among you write equations that explain the wheel operation? Build a water wheel - does the computer model match the real system. Note, depending on what computer you are using, the simulation may run slower than real-time (see pendulum discussion for a definition of real-time). For your reference, the water wheel position is recalculated and redrawn every 0.2 seconds of simulation time.

I'm sure you can come up with lots of things to do with the water wheel. For a good book on chaos, try "Chaos - Making a New Science," by James Gleick. It was published by Penguin Books in 1987.

Parameters:

Other parameters may be changed, in addition to the input flow. These changes are made by clicking **Parameters** on the main screen. Several terms may be modified and each parameter has limits within which it may be set. These parameters and their respective limits are: number of cans (N Cans, 2 to 6), stick length (L Stick, 10 to 300 cm or 4 to 120 in), stick mass (M Stick, 30 to 2300 grams or 1 to 80 ounces), can diameter (D Can, 1 to 100 cm or 0.5 to 40 in), can height (H Can, 1 to 100 cm or 0.5 to 40 in), can mass (M Can, 30 to 2300 grams or 1 to 80 ounces), emptying time (T Empty, 0.1 to 1000 seconds), filling time (T Fill, 0.1 to 1000 seconds), and frictional factor (Friction, 0.00 to 1.00, where 0 is no friction and 1 is maximum friction). To change a parameter, click on the desired field and type the new value. Repeat this for each change. Click **OK** when done changing parameters. You may also select the type of units you desired. Simply make your choice between **US (English)** units (pounds, inches) and **Metric** units (grams, centimeters). Note if you change unit type, current values will be changed to the new units. The changed values will not be exact equivalents of the previous values though due to some adjustments made by the program.

Two parameters need a bit of explaining: the emptying time and filling time. These are needed to establish the input flow maximum and the flow characteristics of the hole in each can. The emptying time is the number of seconds it takes a full can of water to empty through the hole in the bottom. The filling time is the time it takes to completely fill the can with the hole blocked and the input flow at maximum. This time is used to compute the maximum possible input flow. In the program, this maximum flow is divided into twenty segments - each click of the right arrow increases the flow into the cans by one such segment, hence once you have clicked the right arrow 20 times, the flow is at maximum. Likewise, clicking the left arrow will reduce the flow by 1/20th of the maximum.

Help:

Clicking **Help** or pressing **<F1>** brings up this screen of information.

Exit:

Click **Exit** and you are returned to the **Math and Science** main menu screen.

No Sound Card

The **Math and Science** programs will work if your computer is not equipped with a sound card.   However (of course), you will not hear any sounds and some program action may be faster than expected.