

Motion Along a Straight Line, Part II: Acceleration \rightarrow Velocity \rightarrow Position

Introduction

OBJECTIVE: To develop an understanding of the integral relationship between acceleration and velocity, and between velocity and position of an object moving along a straight line.

This module includes two specially designed *Mathematica* commands that animate the integral relations between acceleration and velocity, and between velocity and position of an object moving along a straight line. A variety of motions are studied, including constant velocity, constant acceleration, harmonic oscillation, and decaying oscillations. You can also use the specialized *Mathematica* commands to study other motions that may be of interest to you.

Note: This module uses two specially designed functions,

position[**velocityfunction**_, **timeinterval**_, **initialposition**_], and

velocity[**accelerationfunction**_, **timeinterval**_, **initialvelocity**_].

These are not built-in *Mathematica* functions and are only available in this module. The arguments are a velocity function, **velocityfunction**_, or an acceleration function, **accelerationfunction**_, the time interval over which the motion occurs, **timeinterval**_, and the initial position, **initialposition**_, or the initial velocity, **initialvelocity**_. In the sections that follow you will see examples of how these functions are used.

■ Technology Guidelines

NOTE: If you have just finished a module, restart *Mathematica* or close the *Kernel* before executing a new module.

TO OPEN CELLS, put your cursor on the right cell bracket and double click.

INITIALIZATION CELLS

When asked if you want to "... automatically evaluate all the initialization cells in the notebook ...," respond by pressing the "Yes" button.

TO STOP AN EXECUTION

Select the *Kernel* pull-down menu and click on *Abort Evaluation*.

ORDER OF EXECUTION

Execute cells in the order given. Do not skip any Input cells within a given notebook.

SAVING NOTEBOOKS

You can save anytime to any directory you choose, and it is wise to save often.

However, before you do your final save, delete all your output by selecting the

Delete All Output selection under the *Kernel* pull-down menu.

EXPERIENCING MAJOR PROBLEMS

Save if appropriate, then shut down *Mathematica* and start it up again.

Part I: Constant Velocity


To measure position, select a reference point or origin on the line and scale the line in appropriate units (e.g., meters, feet, or miles). A distance and a direction specify the position. One direction from the reference point along the reference line is taken as positive and the other direction as negative. The distance is measured on the scaled line.

Suppose that you drive your car along a straight road at a constant speed of 60 mph for 4 hours and then turn around and come back at the same speed. Let's take the starting point to be the origin and the direction your car travels during the first four hours as positive. The velocity is given by the following function.

In[18]:=

```
Clear[v];

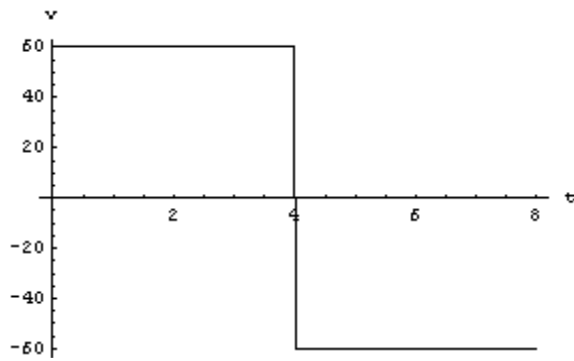
v[t_] = Which[t ≤ 4, 60, t > 4, -60];

">  About Mathematica
```

When the velocity is positive, you are moving in the direction you selected to be positive, and when the velocity is negative, you are moving in the negative reference direction. The magnitude or absolute value of the velocity is the speed of the object, in this case, your car. Now we plot the velocity function for the duration of the trip.

In[20]:=

```
Plot[v[t], {t, 0, 8}, AxesLabel → {"t", "v"}];
```



">  *About Mathematica*

You may obtain your position at any specific time by integrating the velocity from the time at the beginning of the trip up to an arbitrary time t , that is,

$$s(t) = \int_{t_0}^t v(u) du + s(t_0).$$

Assuming that $t_0 = 0$ and $s(0) = 0$, let's find the position function and plot it.

In[21]:=

```
Clear[s, s0];
```

```
s0 = 0;
```

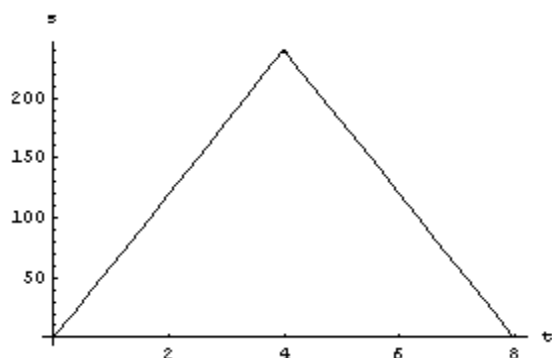
```
s[t_] = Integrate[v[u], {u, 0, t}] + s0
```

Out[23]=

$$\int_0^t \text{Which}[u \leq 4, 60, u > 4, -60] du$$

In[24]:=

```
Plot[s[t], {t, 0, 8}, AxesLabel -> {"t", "s"}];
```



Like velocity, position has a size or magnitude and a direction. The magnitude of the position is the distance from the reference point (in this case the starting position). The sign of the position function indicates a direction, for example, to the right or left of the reference point. If the velocity function, $v(t)$, is positive, then the position function is increasing with time, and if it is negative, then the position function is decreasing as time progresses. Mathematical quantities that have a magnitude and a direction are called vectors. The position and velocity of a moving object are both vector quantities.

The position at time t is the accumulation or sum of infinitesimal displacements from the starting time, added to the initial position, that is,

$$s(t) = s(t_0) + \int_{t_0}^t ds = s(t_0) + \int_{t_0}^t \frac{ds}{dt} dt = s(t_0) + \int_{t_0}^t v dt.$$

The specially designed **position[]** command in the input cell that follows illustrates this idea by producing a sequence of graphs that show the accumulation of displacements in two different formats. The first format shows the displacements as shaded areas between the velocity function and the t -axis, and the second format shows them as a change in the position of the object on the position graph.

Red areas on the velocity graph represent increases in the position due to positive velocity; blue areas represent decreases in position due to negative velocity. Similarly, the red arrows on the position graph represent increases in the position function, and the blue arrows represent decreases. The lengths of the red arrows on the position function are numerically equal to the corresponding red areas on the velocity graph, and the lengths of the blue arrows on the position function are numerically equal to the corresponding blue areas on the velocity graph.

To animate the sequence of graphs generated by the **position[]** command, do the following:

1. If necessary, widen the notebook window so that both graphs in each cell show across the page.
2. Put the cursor in the cell bracket that contains all the graphics cells and double click the left mouse button. This will collapse all the graphs into one cell, displaying only the first graphics

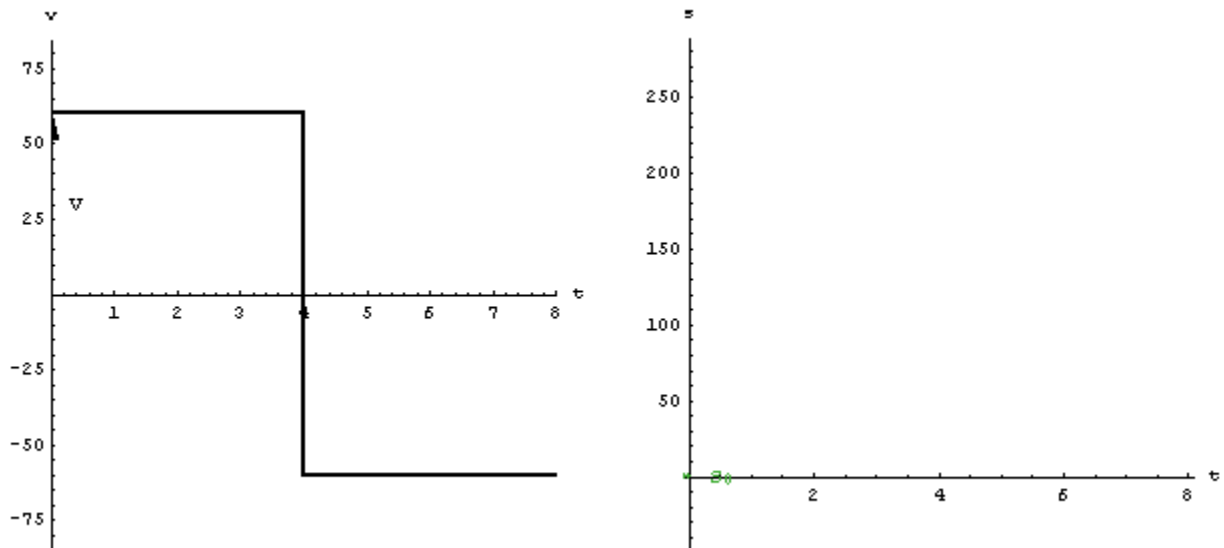
cell in the sequence.

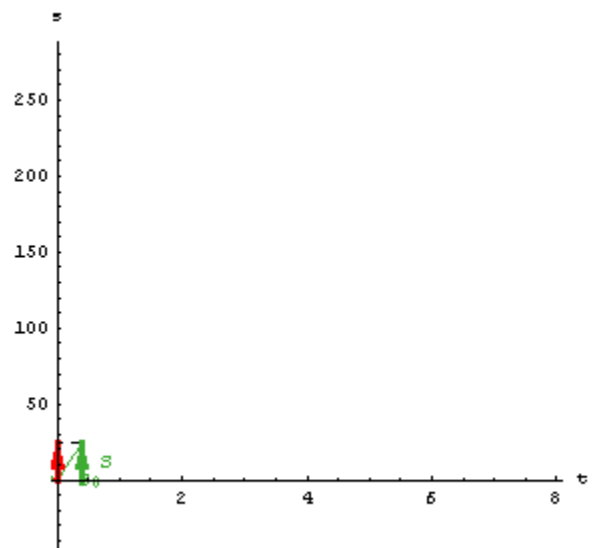
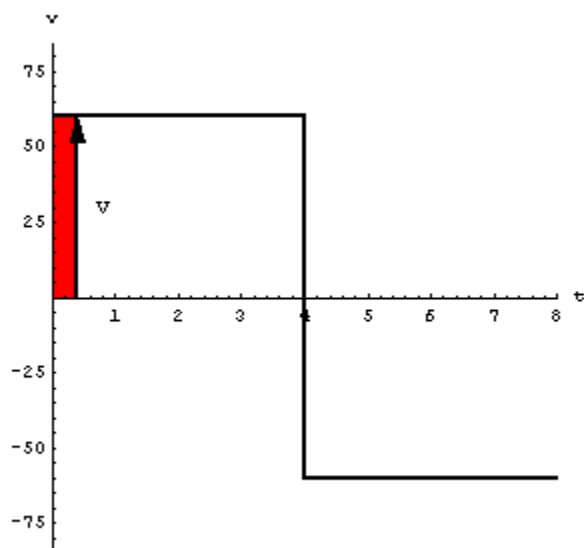
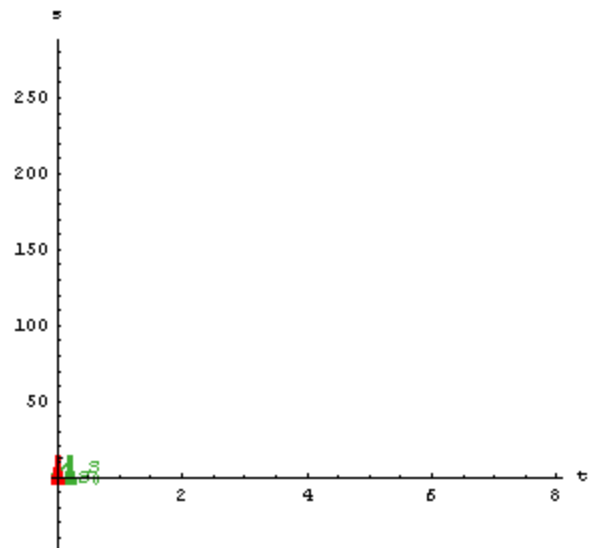
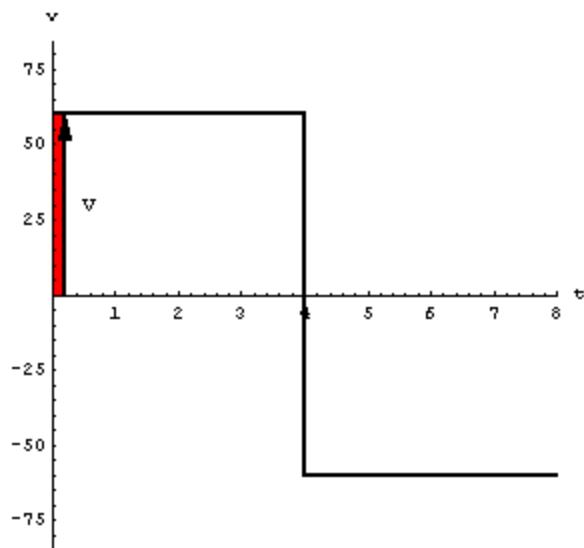
3. Be sure the cell bracket that contains the collapsed graphics cells is selected (if it is not, place the cursor in the cell bracket and click once), and then press Ctrl+Y. This will play the sequence of graphics slides to generate the animation.
4. While the animation is playing, a control bar appears at the bottom of the notebook window. This bar allows you to control the speed and direction of the animation.

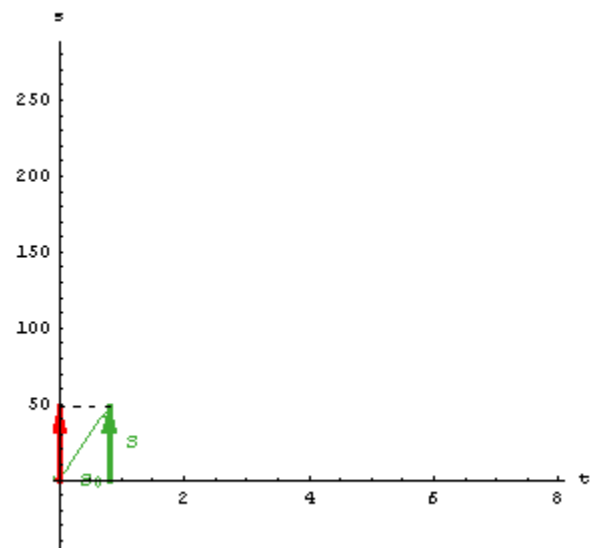
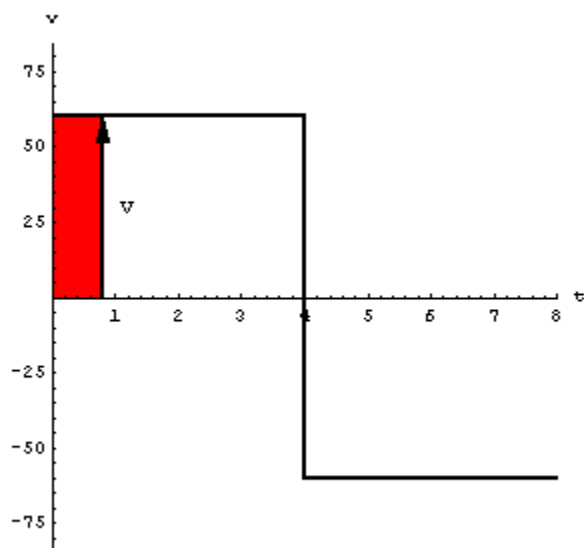
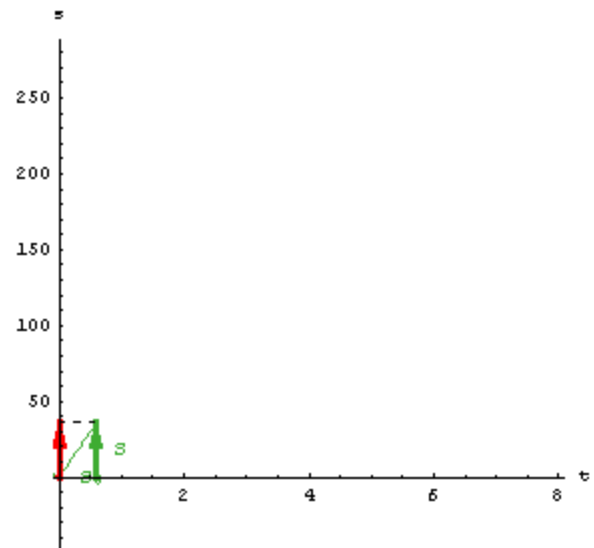
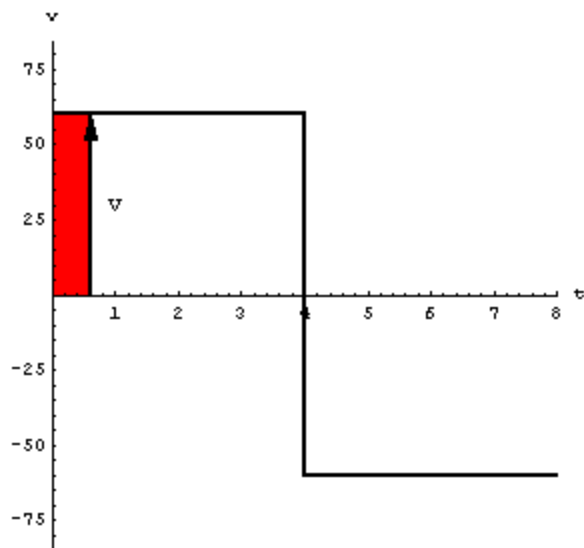
Since the following command generates a lot of output, you probably won't want to print this notebook until after you have deleted most or all of the graphs. We recommend deleting all but a few representative cells from the sequence before you print.

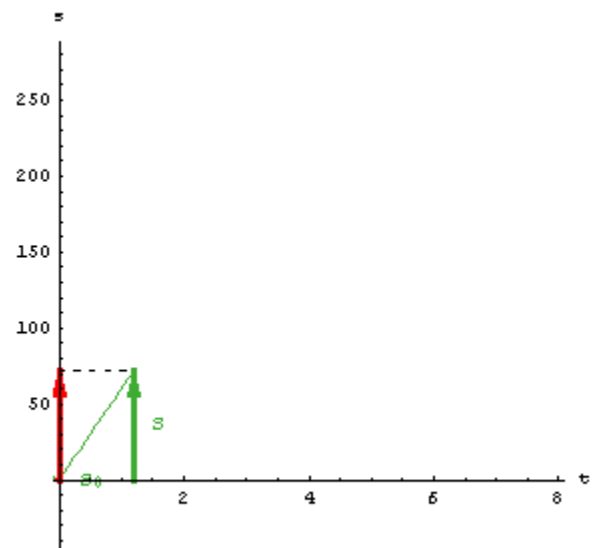
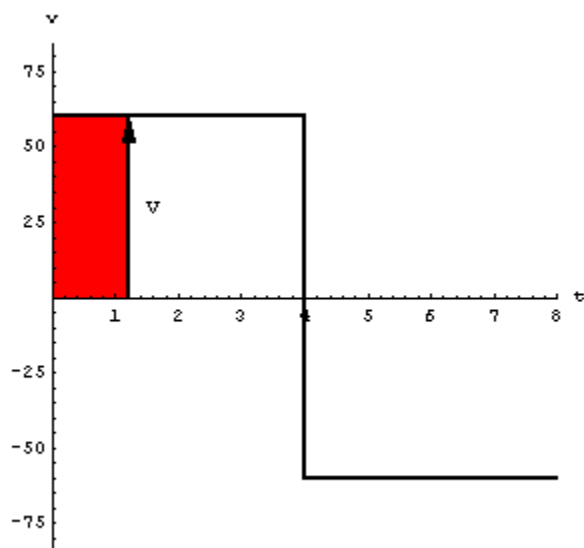
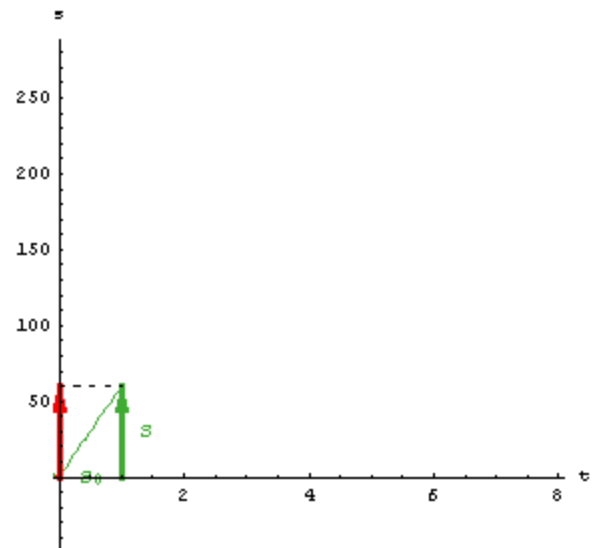
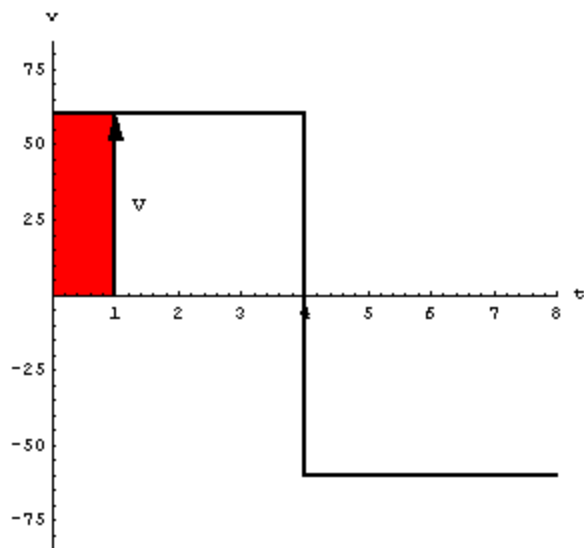
In[25]:=

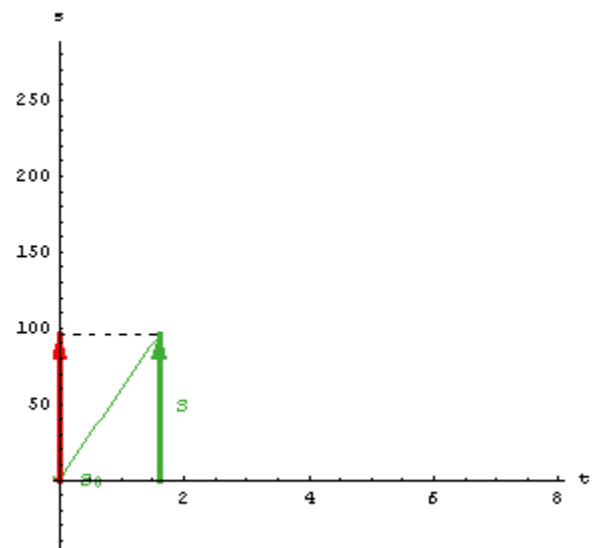
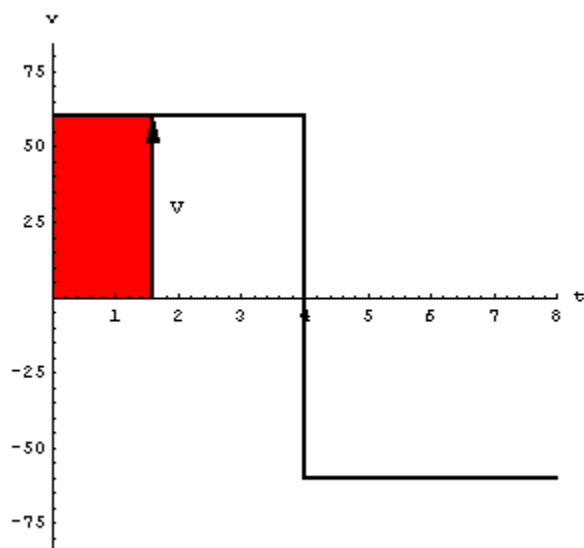
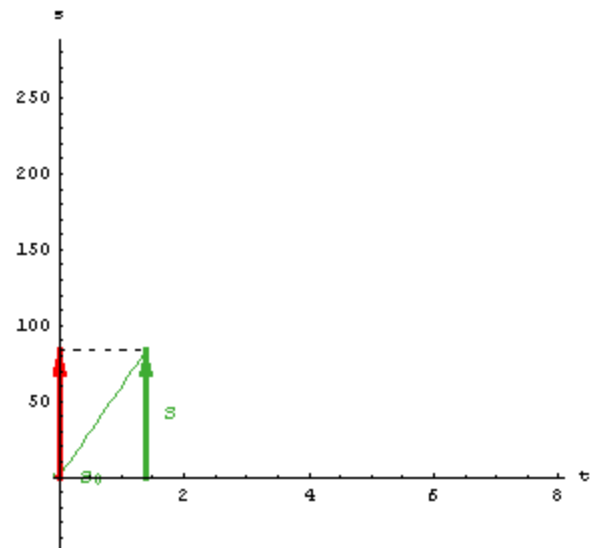
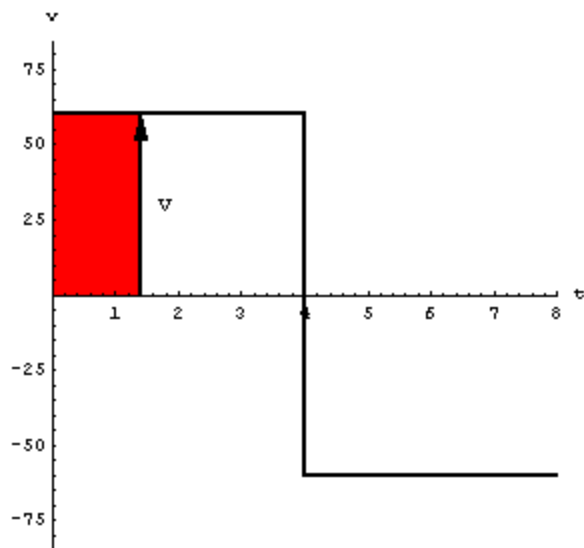
```
position[v[t], {t, 0, 8}, s0]:
```

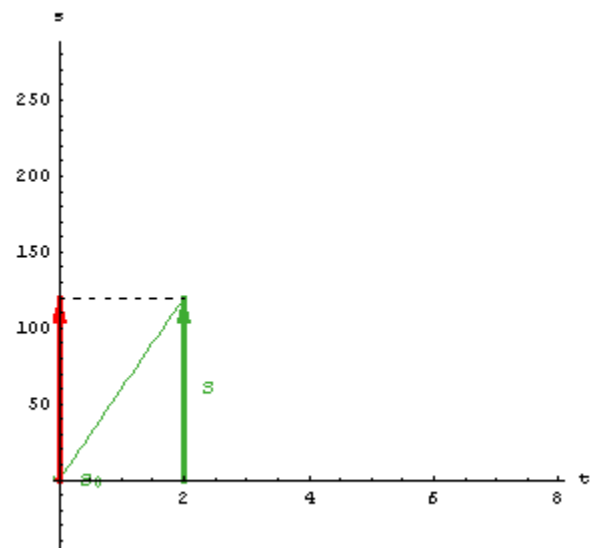
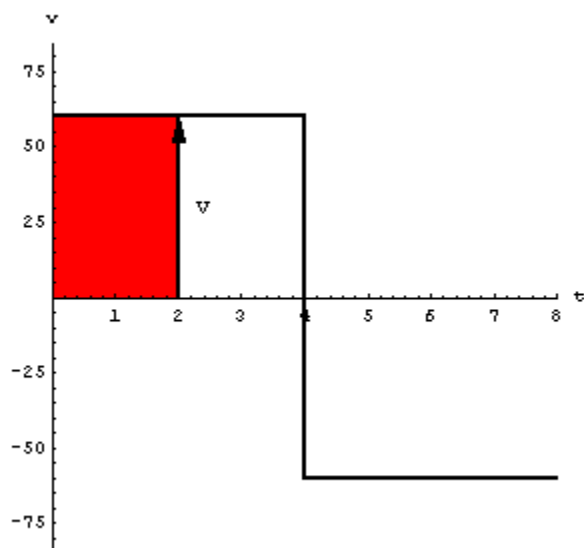
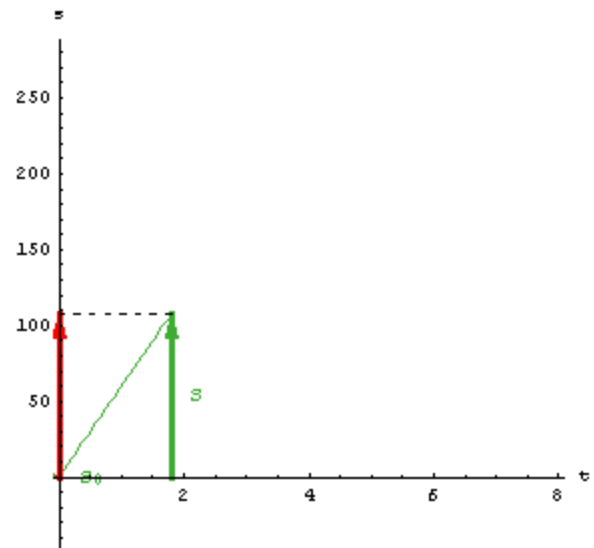
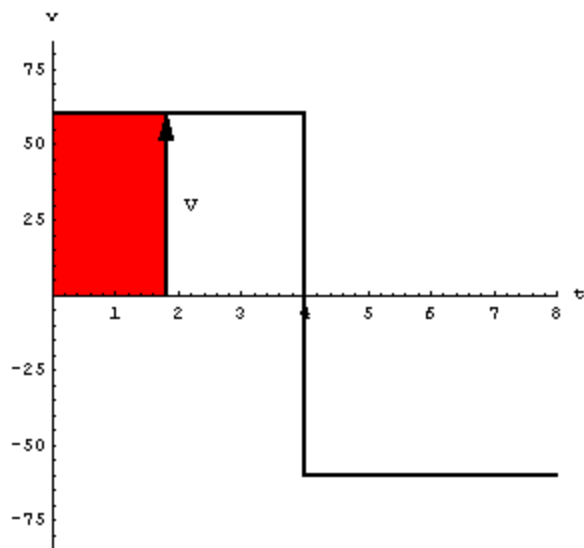


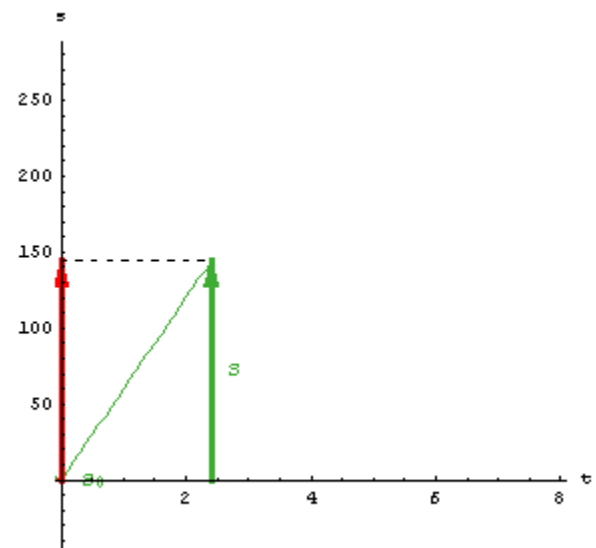
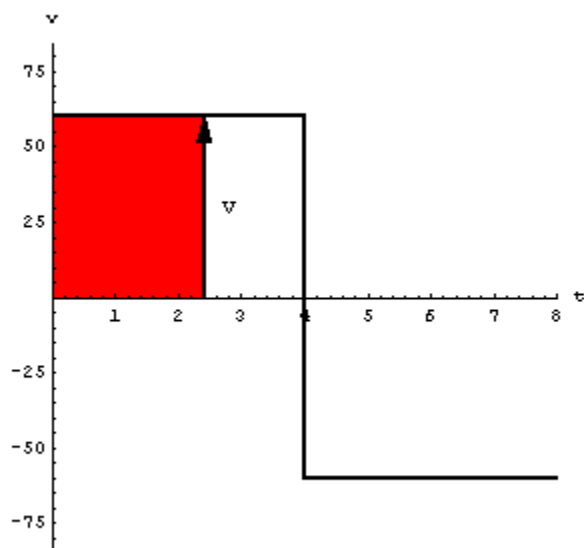
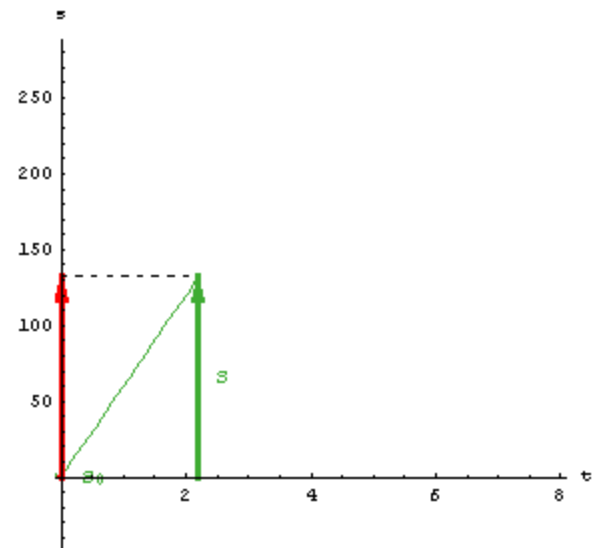
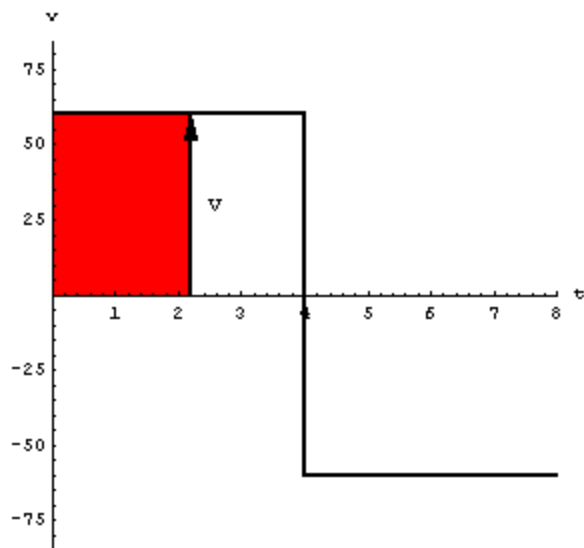


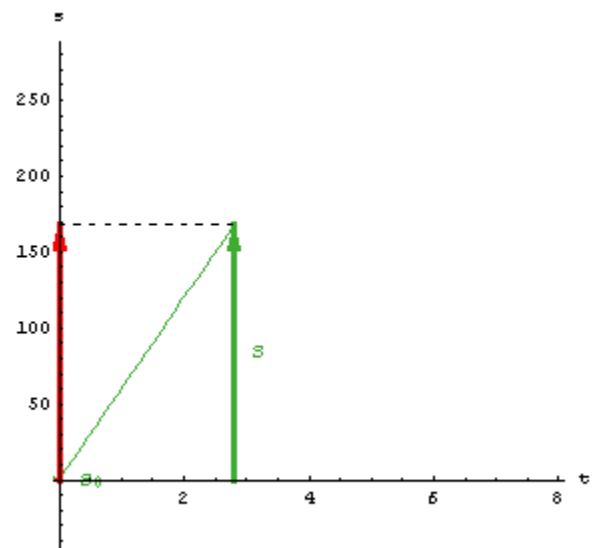
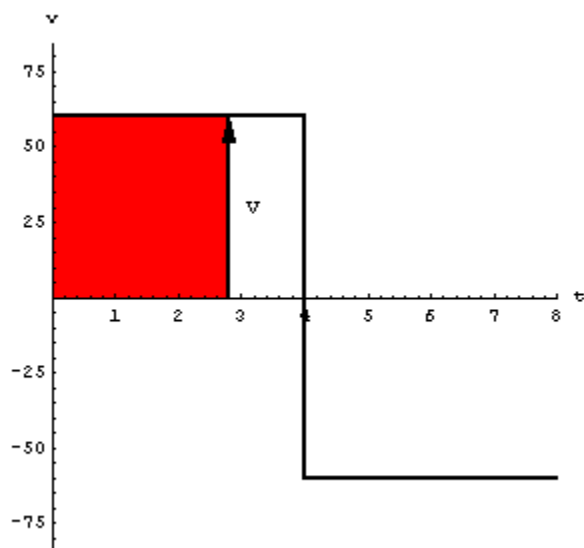
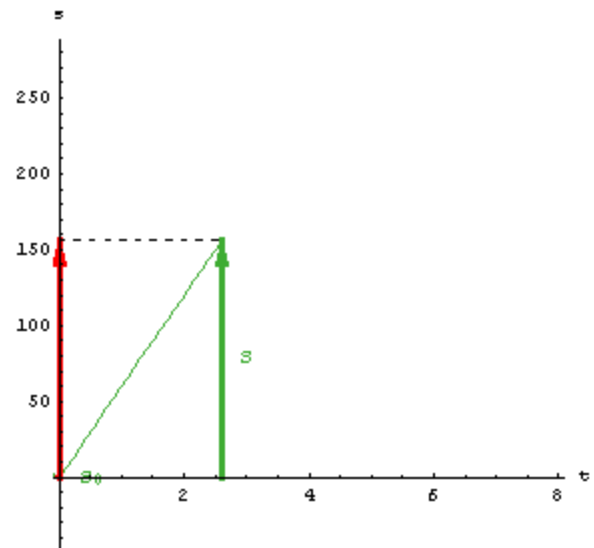
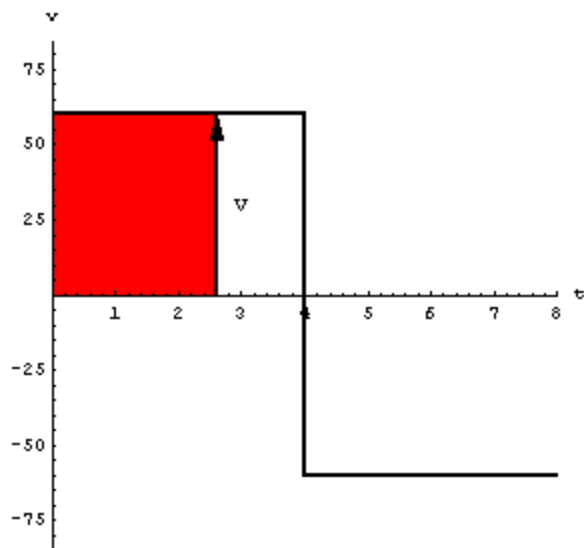


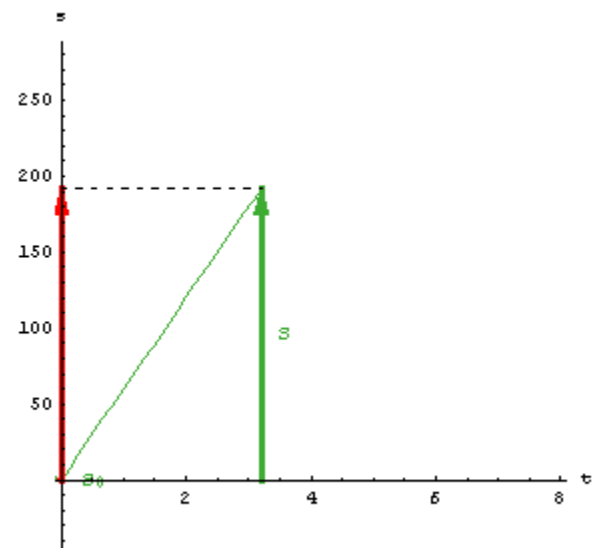
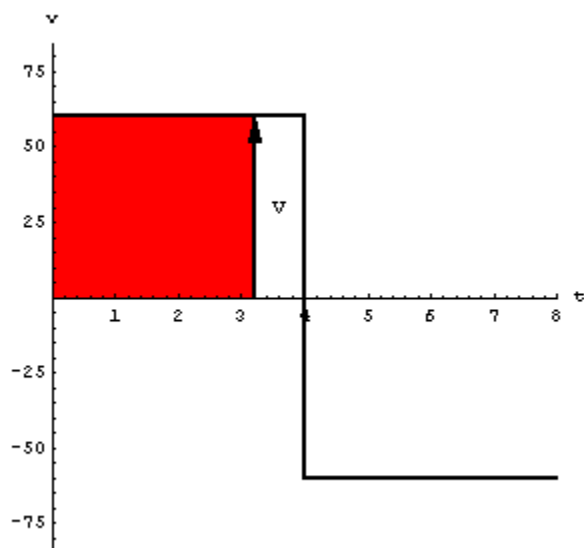
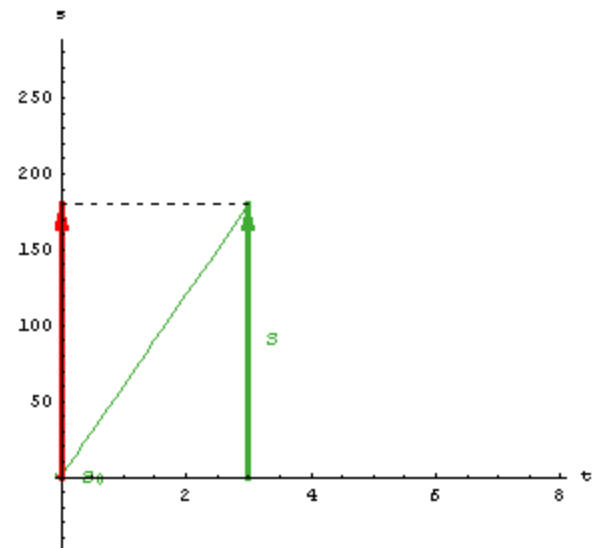
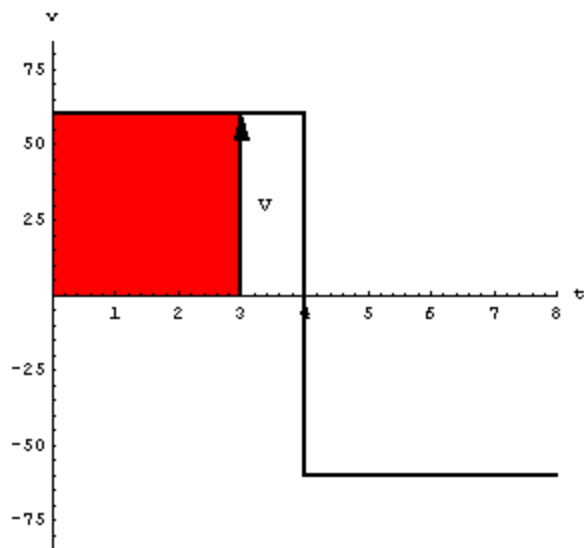


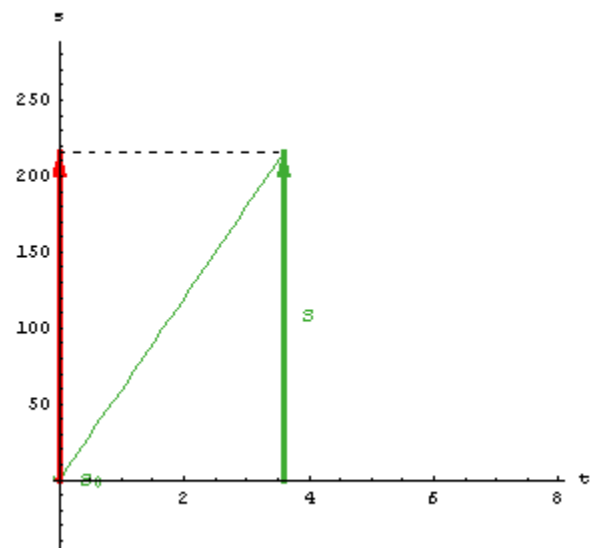
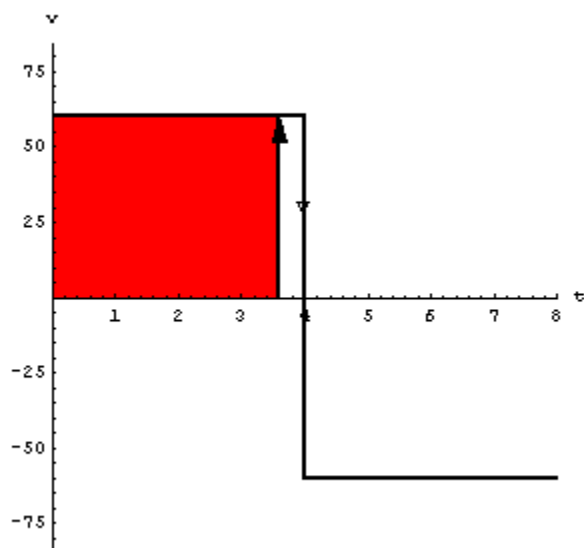
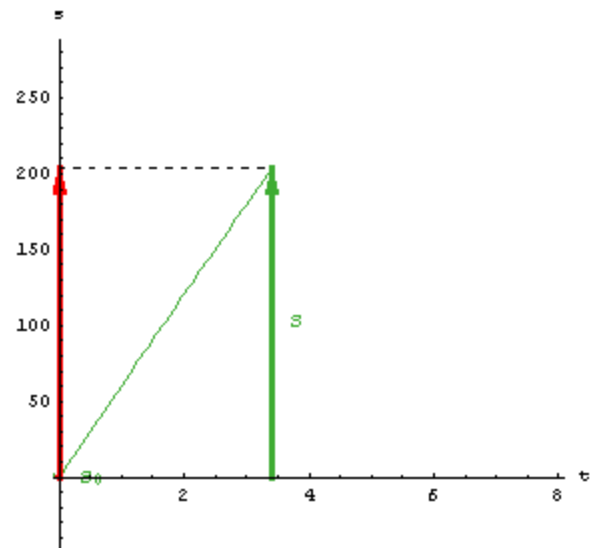
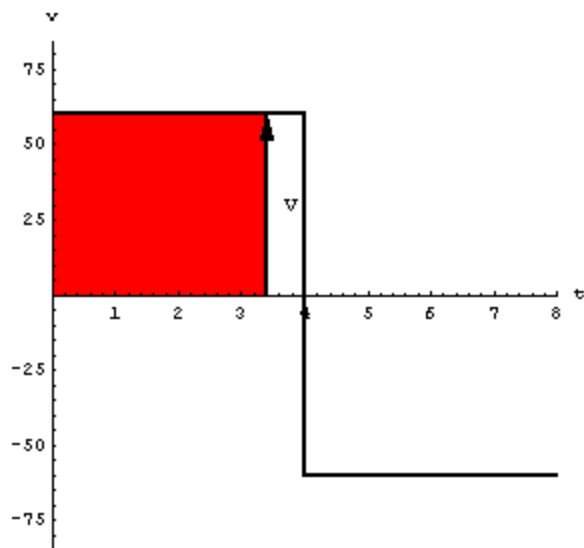


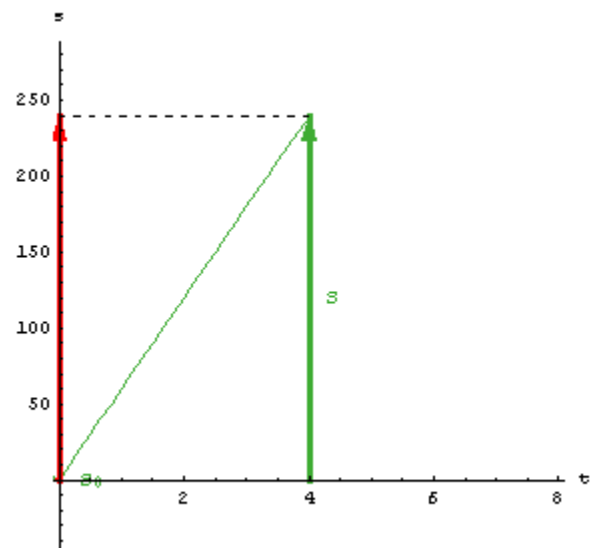
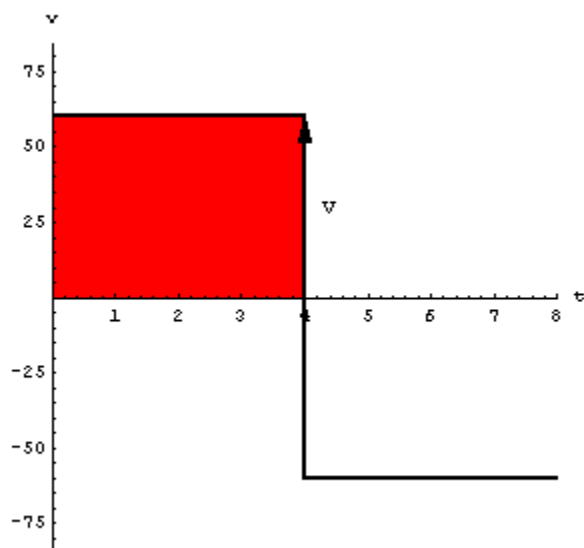
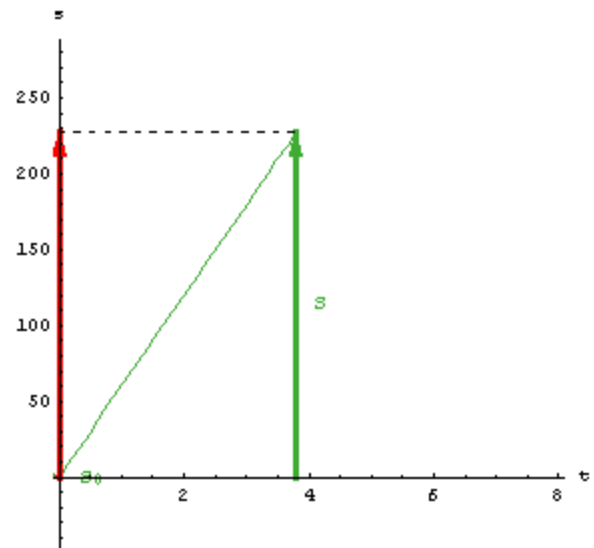
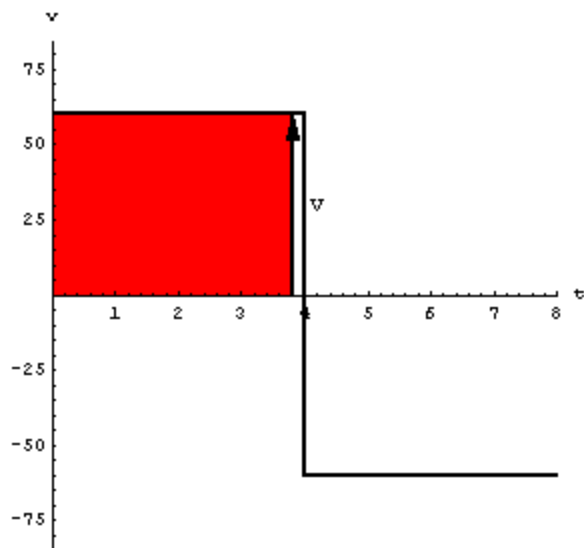


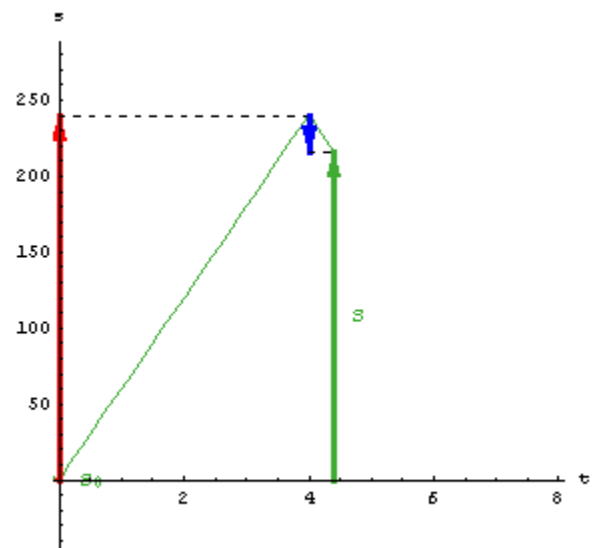
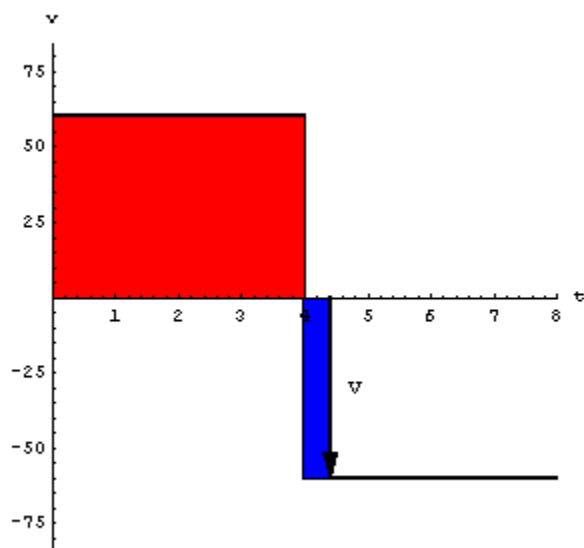
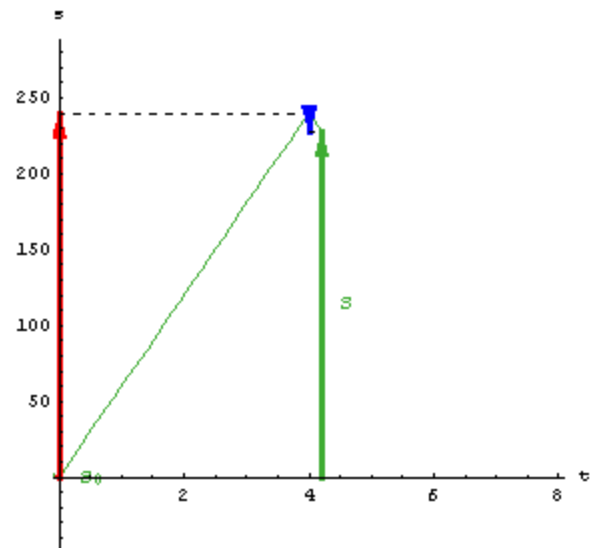
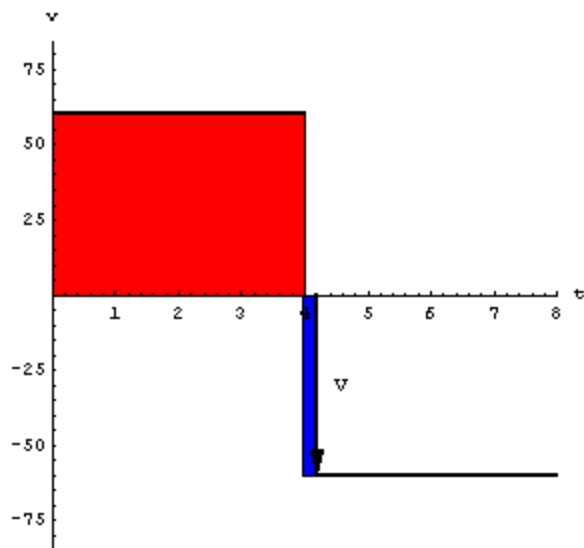


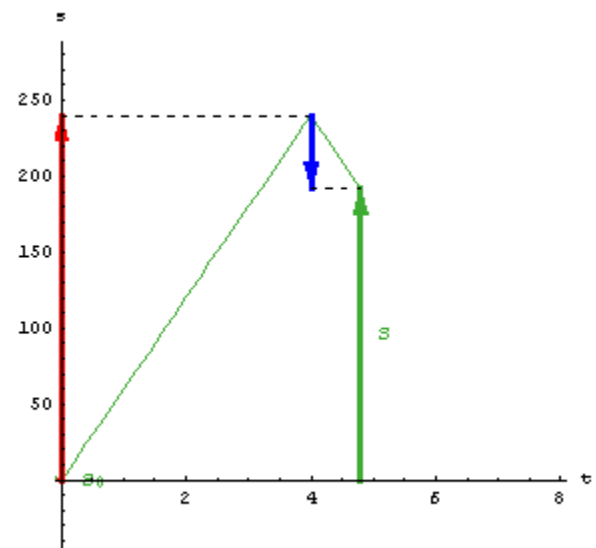
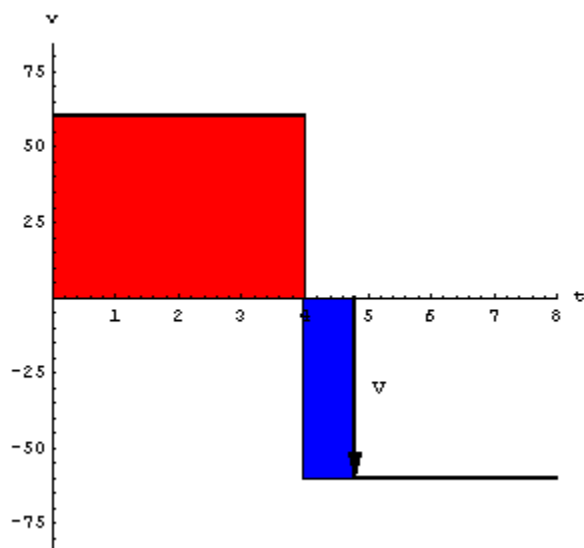
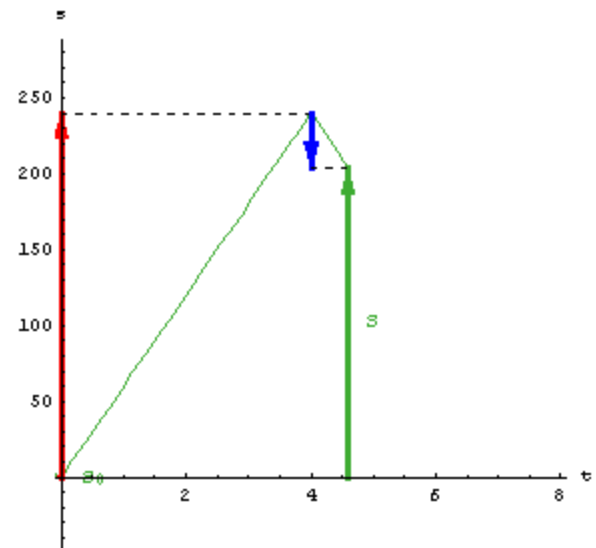
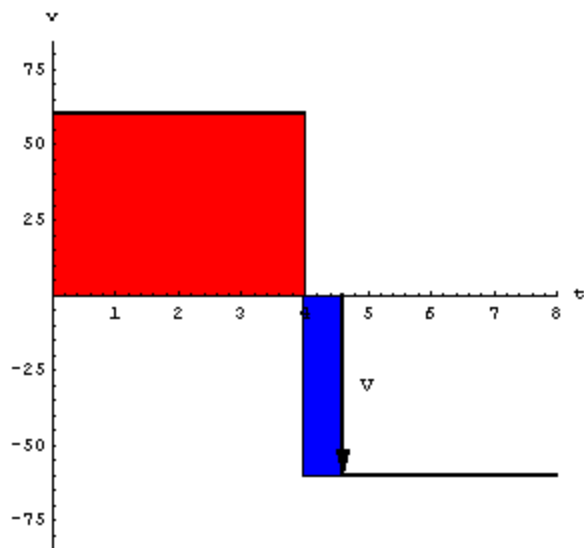


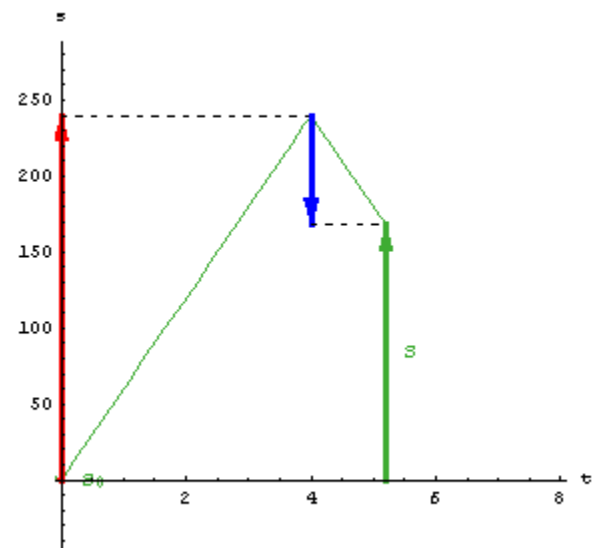
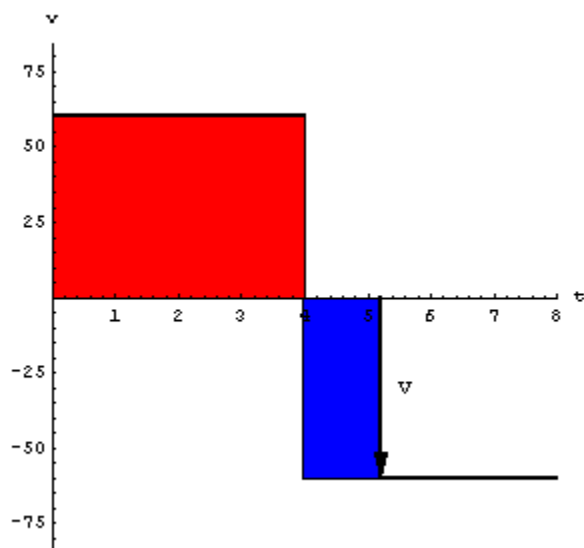
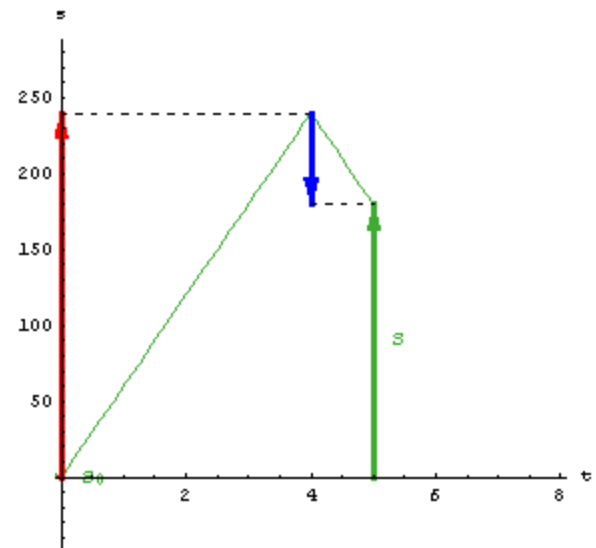
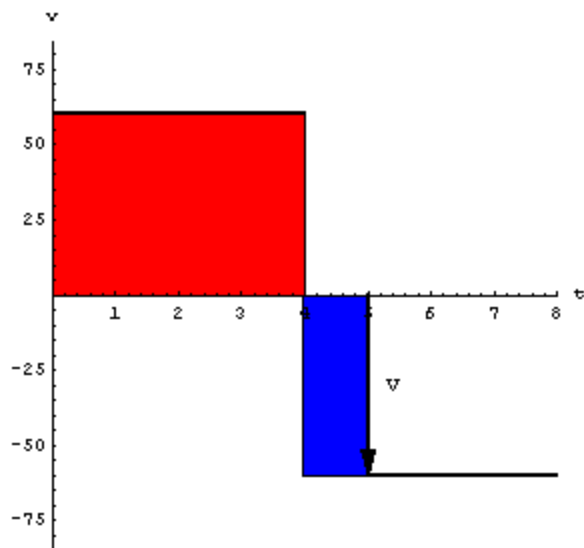


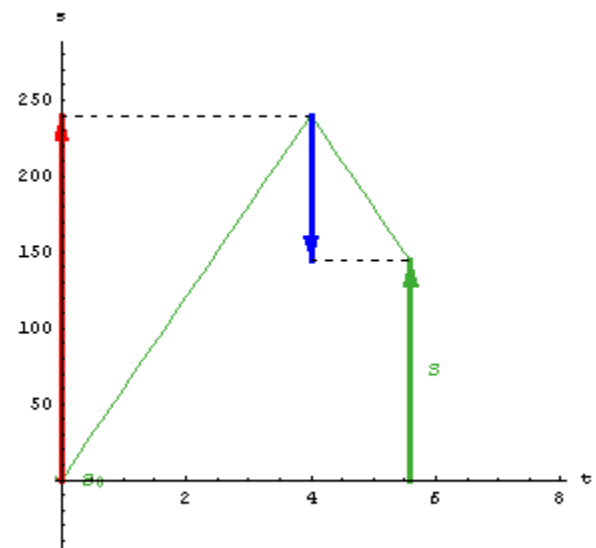
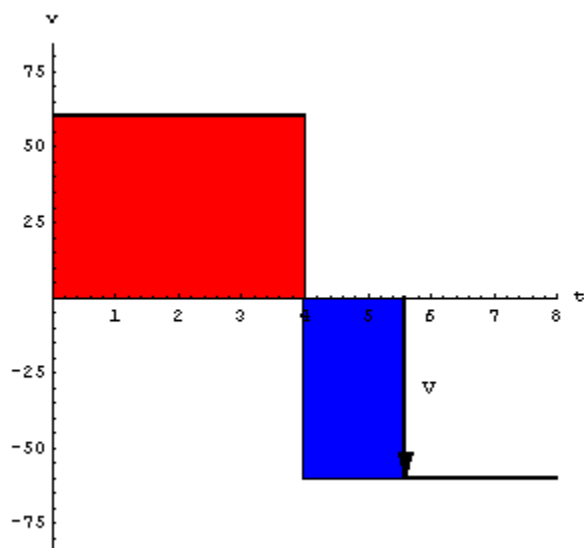
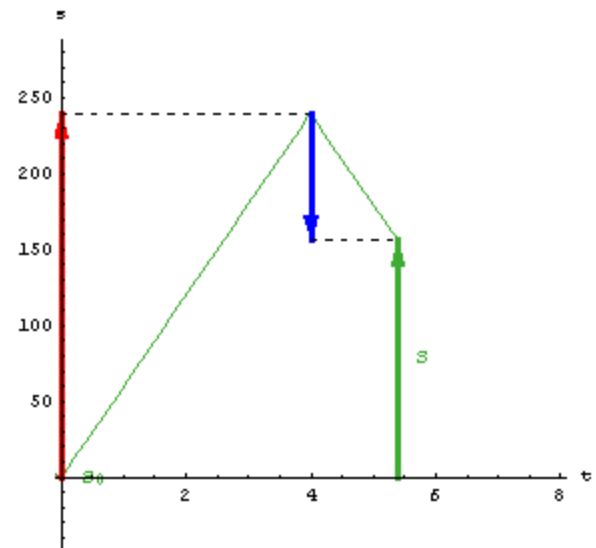
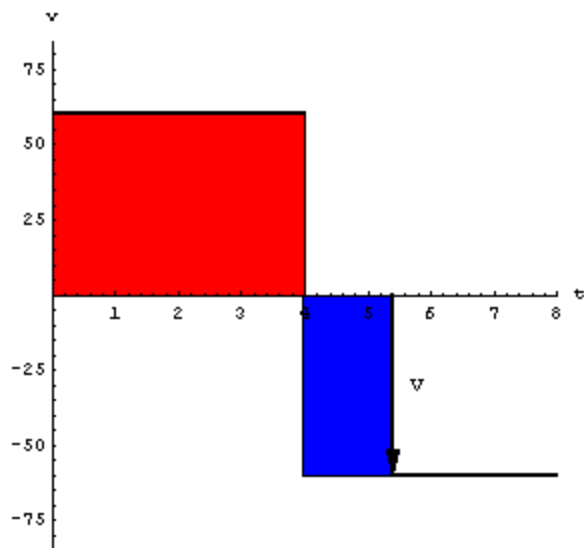


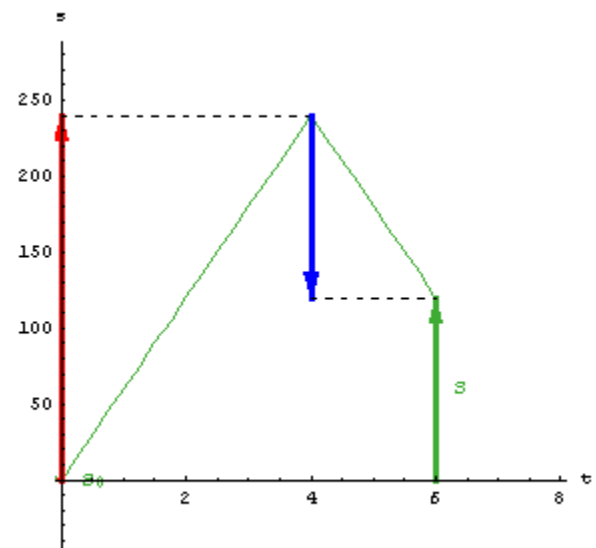
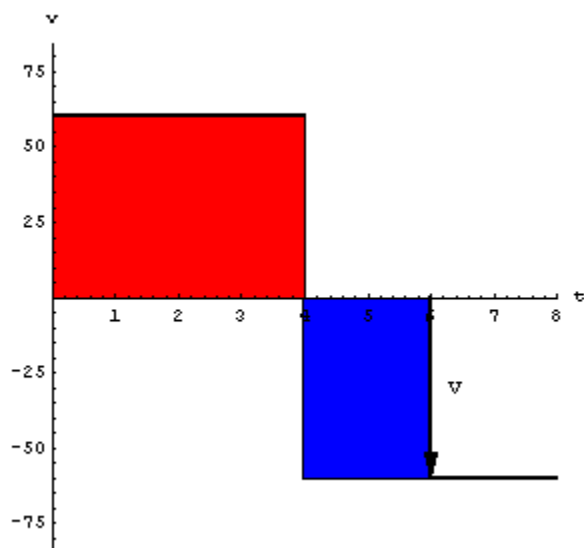
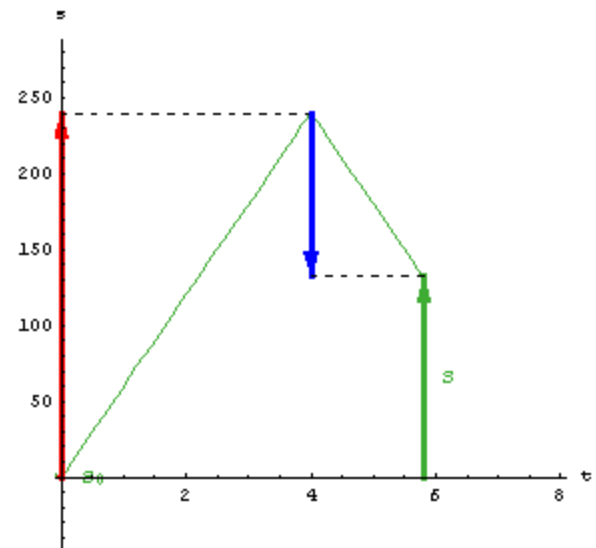
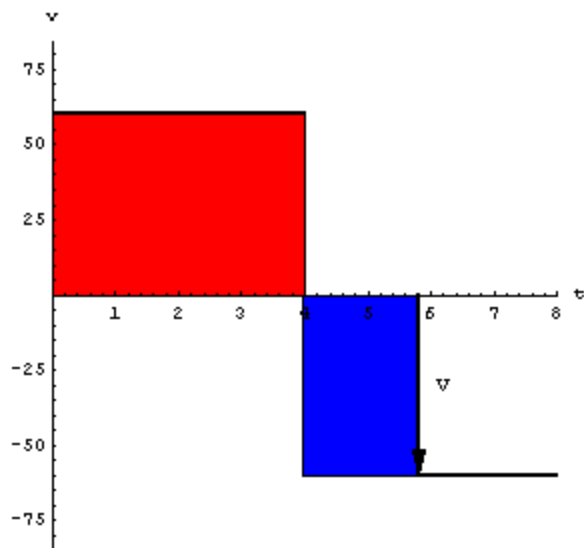


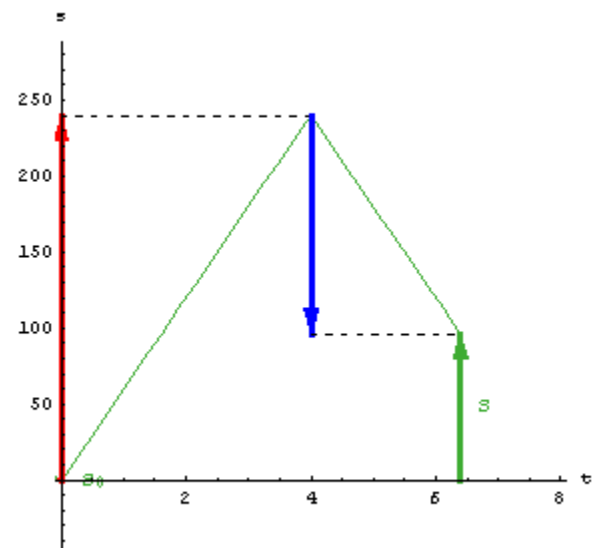
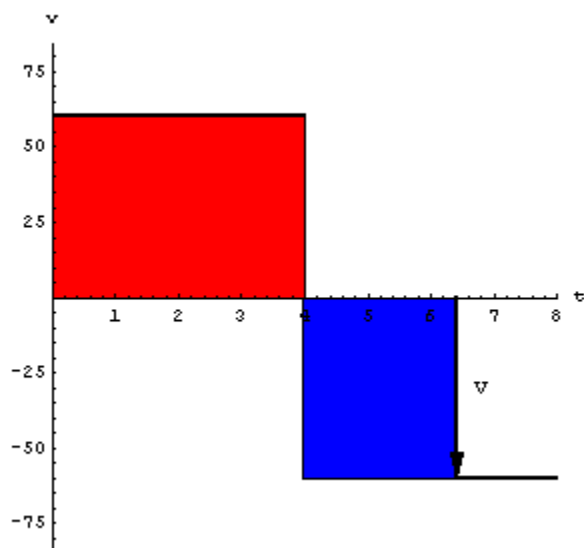
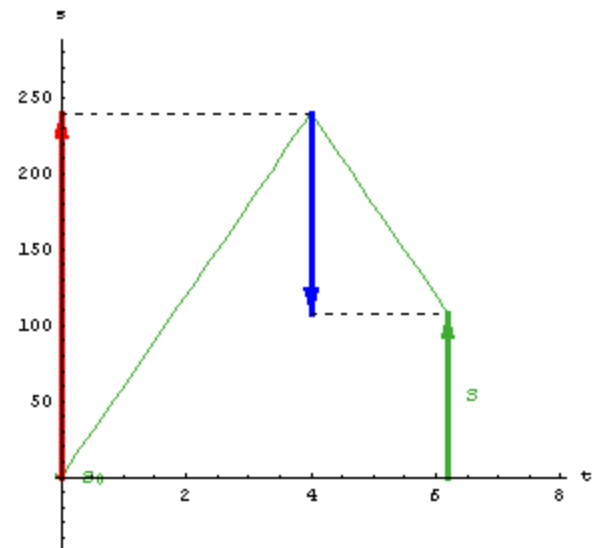
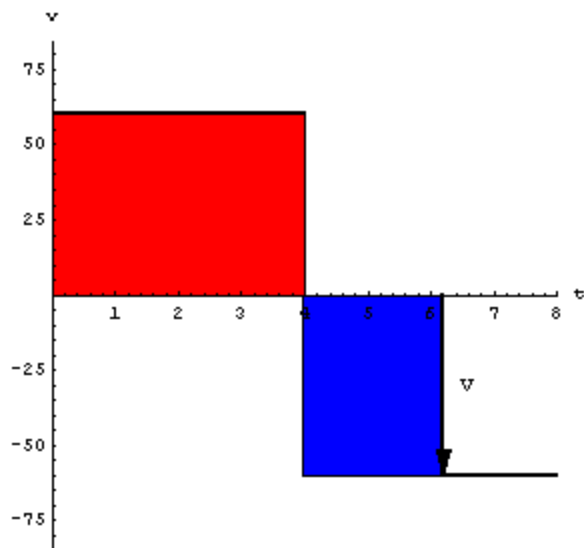


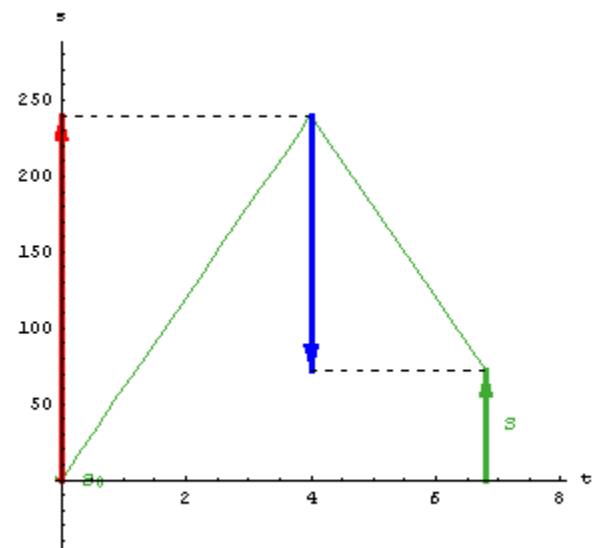
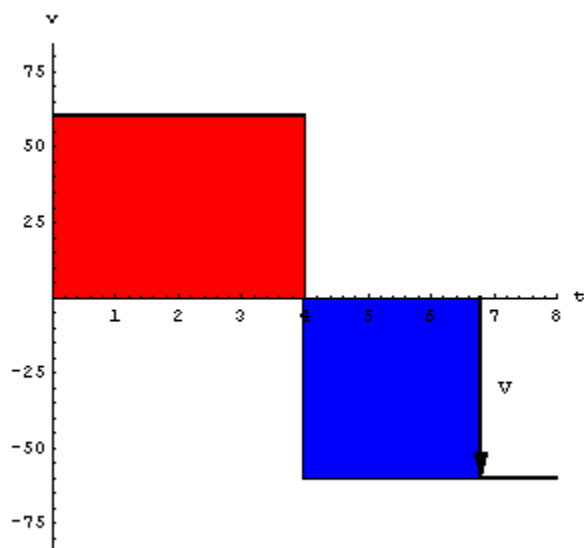
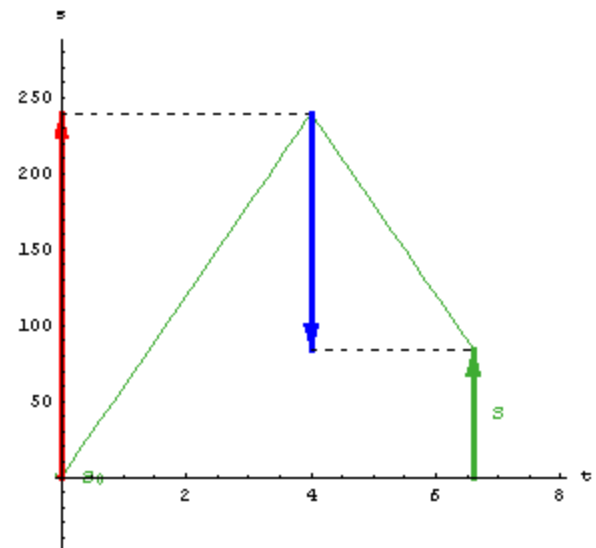
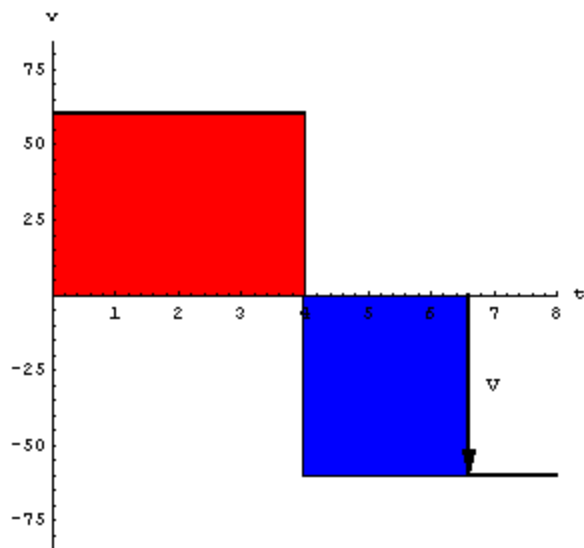


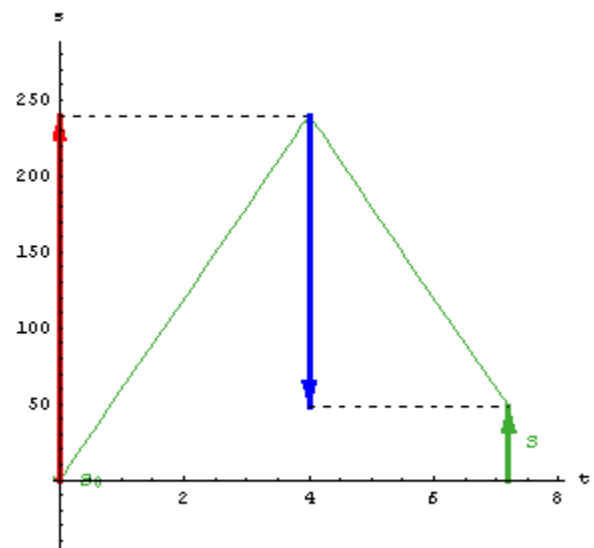
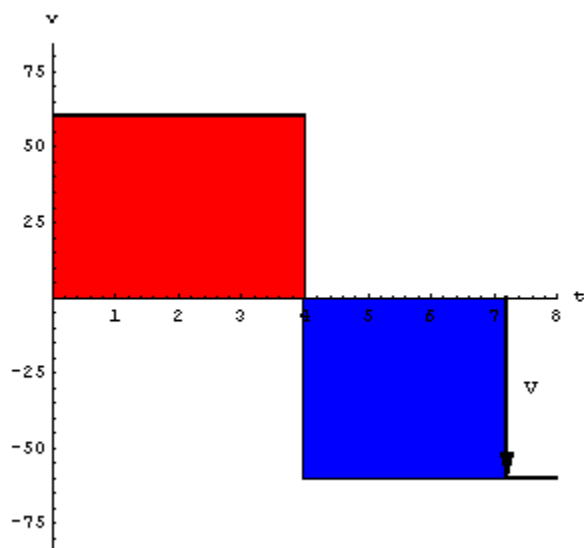
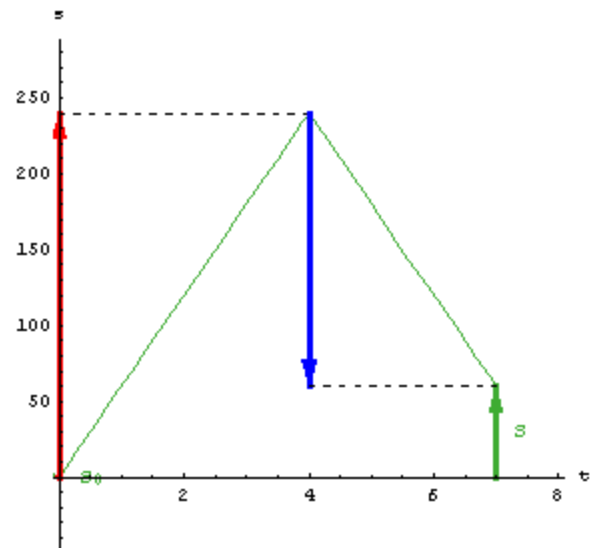
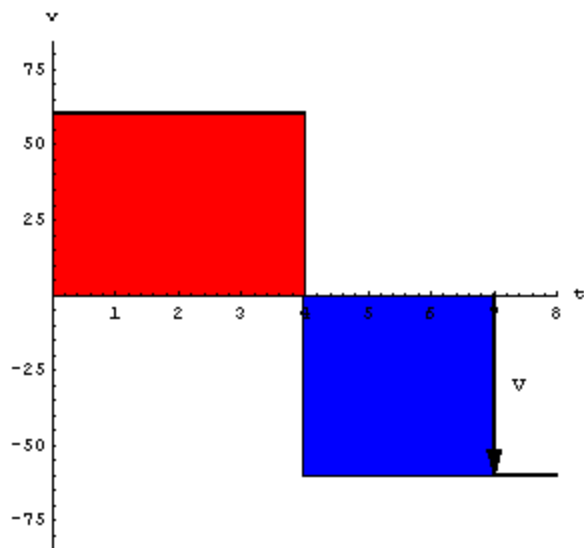


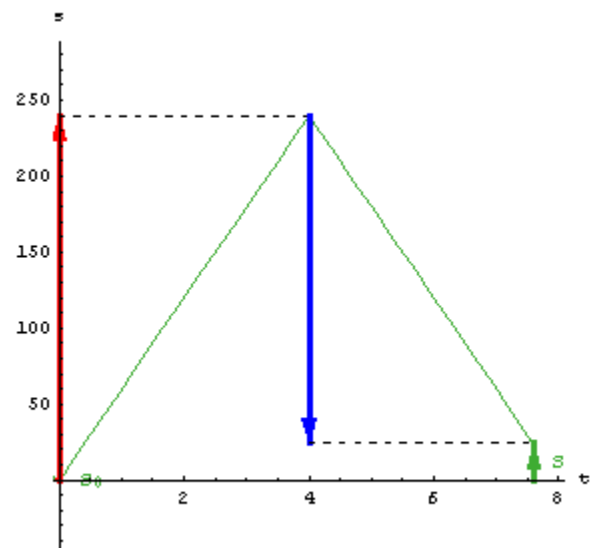
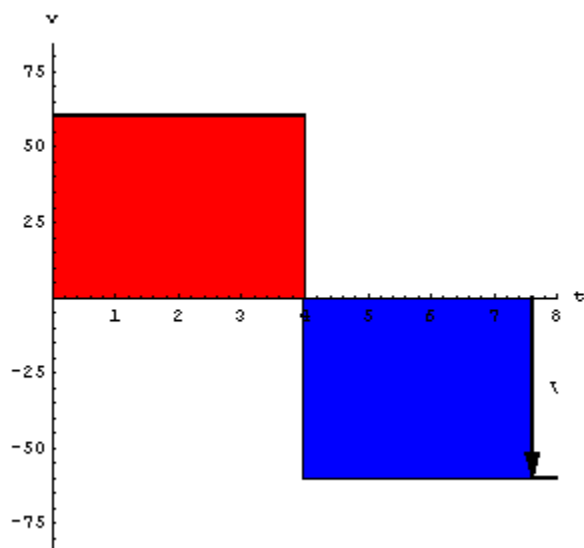
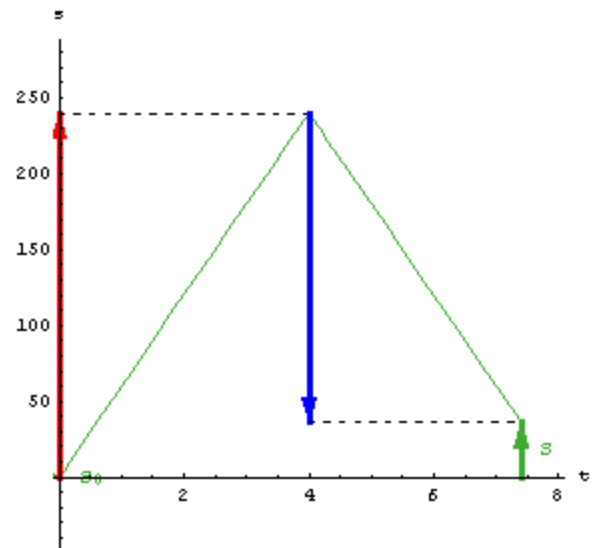
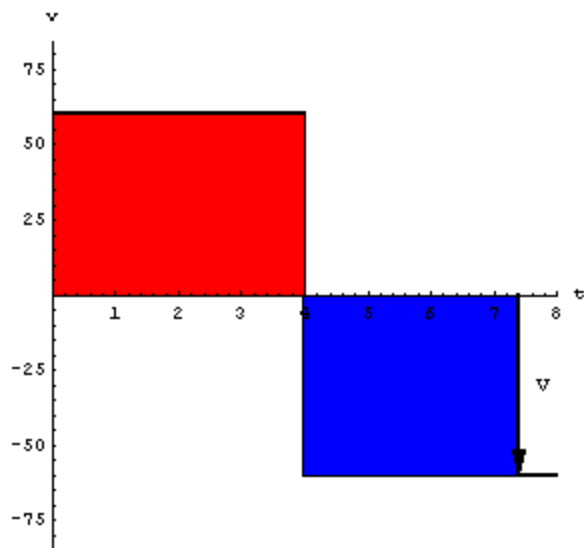


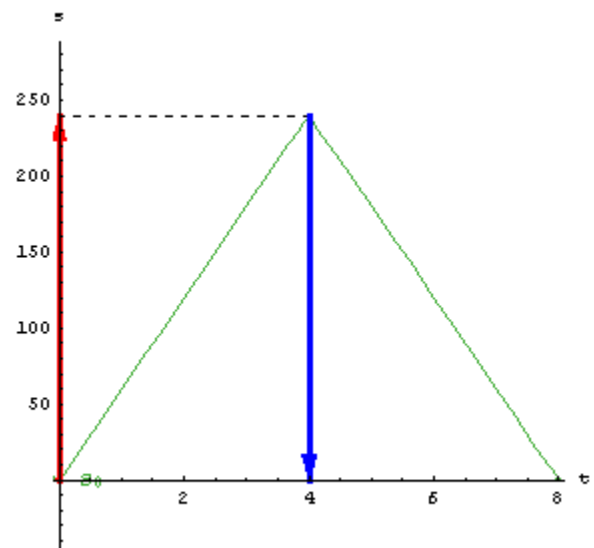
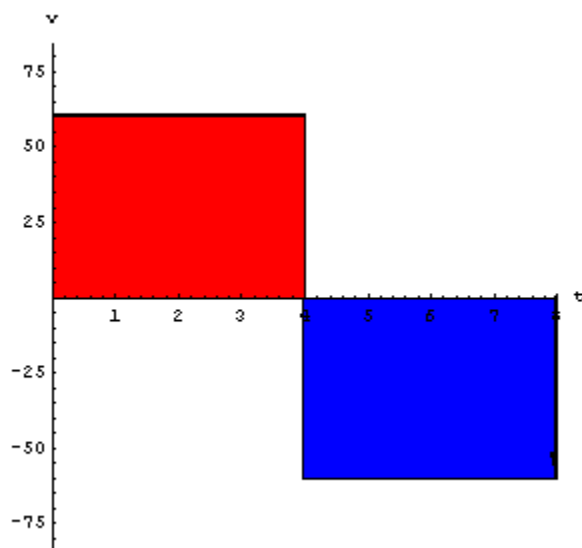
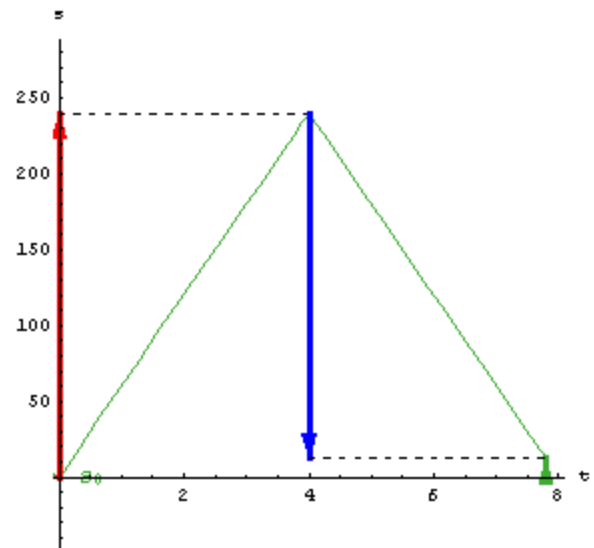
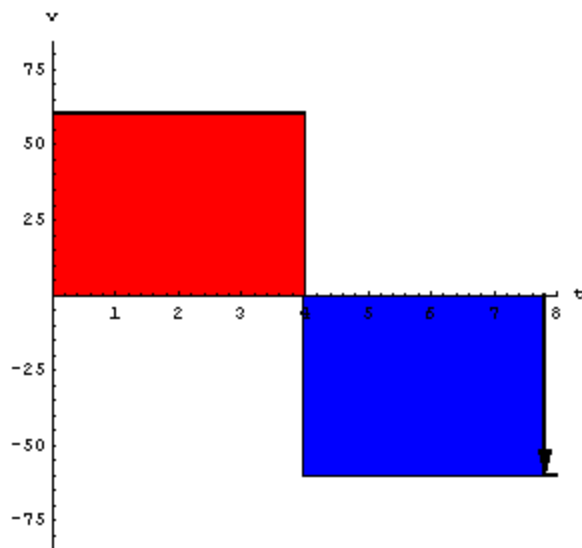












A note about modeling: In the model above, the velocity is undefined at $t=4$ hours. In reality, however, your car would have to slow down, turn around, and accelerate back to cruising speed at $t=4$ hours. This maneuver would occur over a very short interval of time (say a few minutes) when compared to the duration of the trip. The sharp corner at the turnaround time would in actuality be rounded, and the sudden jump from 60 mph to - 60 mph would actually have a finite slope to it. On the scale of eight hours, however, the graphs would look much like those depicted above, even if the more precise model were used. With these observations in mind, you may claim that the above model is reasonable for describing your trip.

You Try It: Different Starting Positions, Directions, and Speed

1. Try different starting positions by changing the value of s_0 , which is highlighted in red in the cell that follows. We suggest that you try $s_0 = -100$ and $s_0 = 100$. What do you observe about the graphs of $s(t)$?
2. Suppose you take a different trip, where you go in the opposite direction for 5 hours at 65 mph, and then return to the start position at the same speed. Use the same reference directions as in Part I and change the velocity function entries that are highlighted in blue, accordingly. If you wish, try different starting values as in the first exercise.

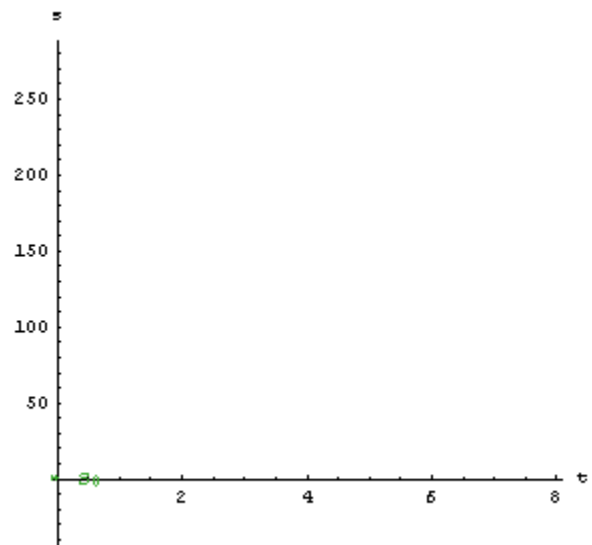
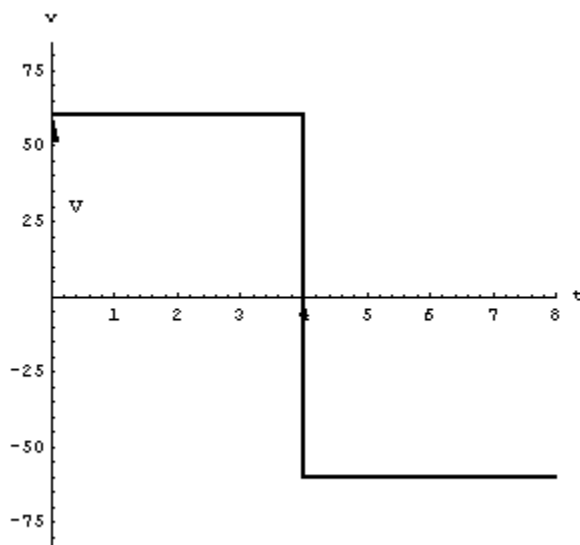
In[26]:=

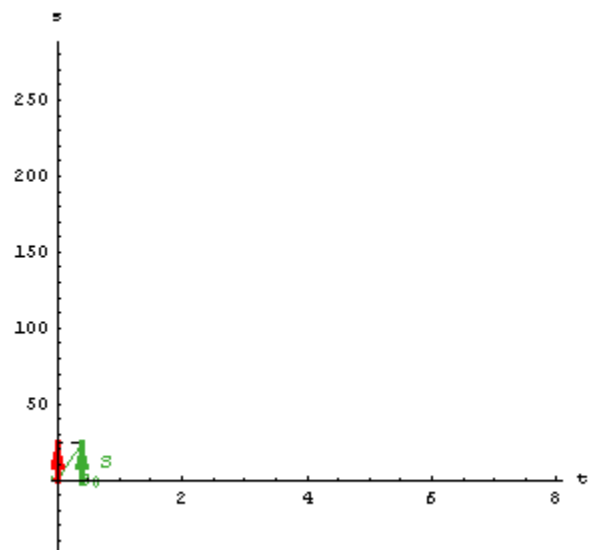
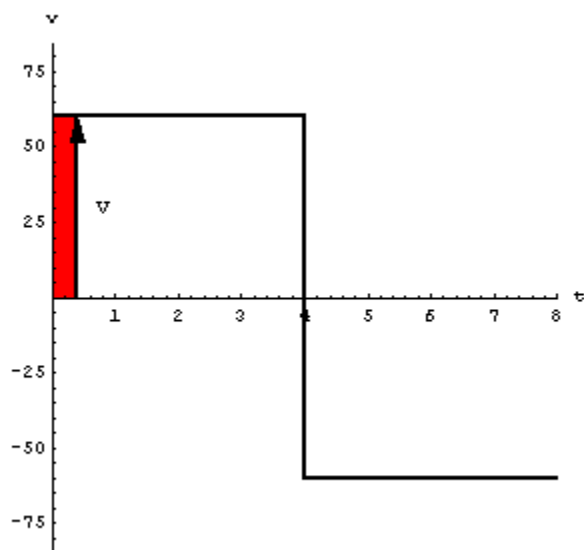
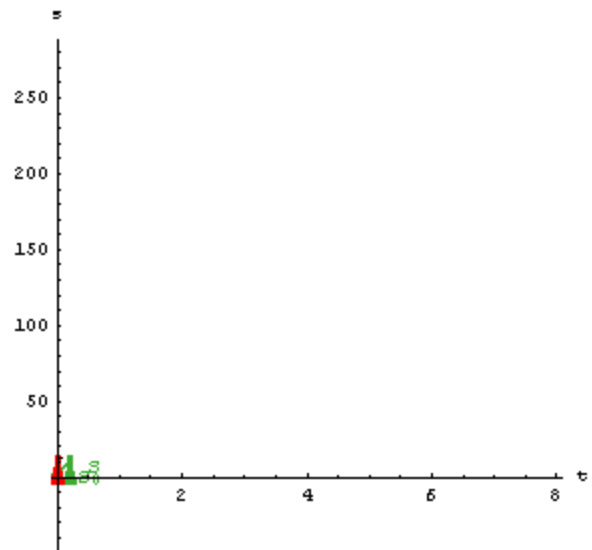
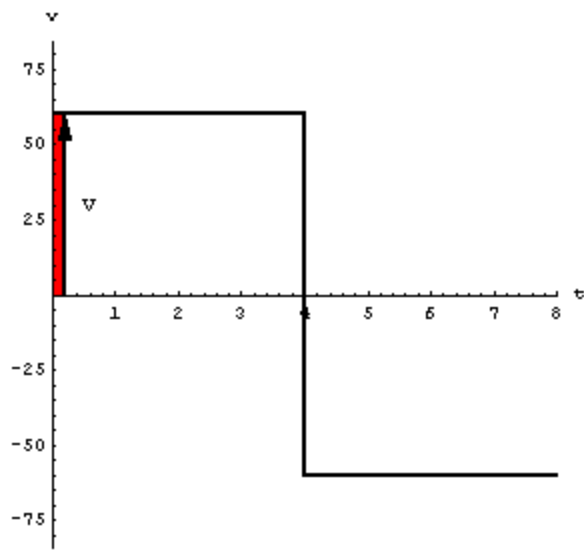
```
Clear[v, s, s0];
```

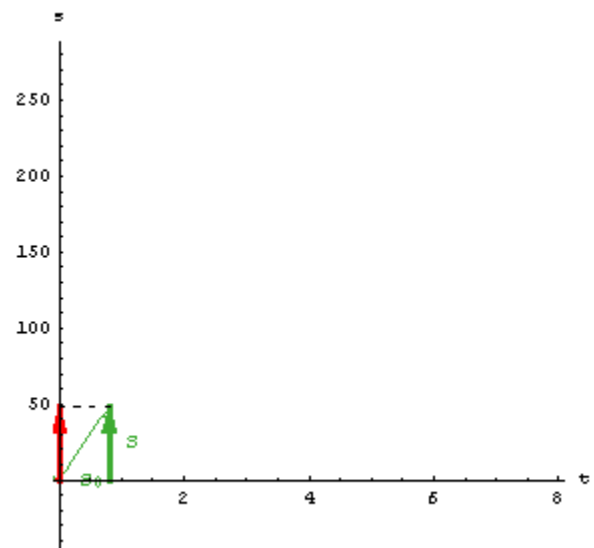
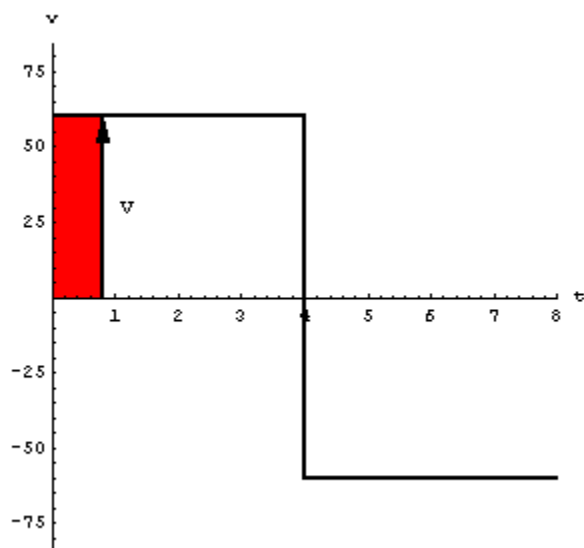
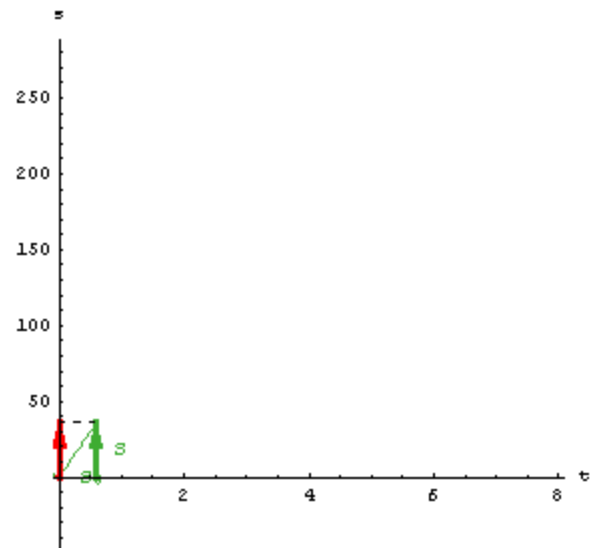
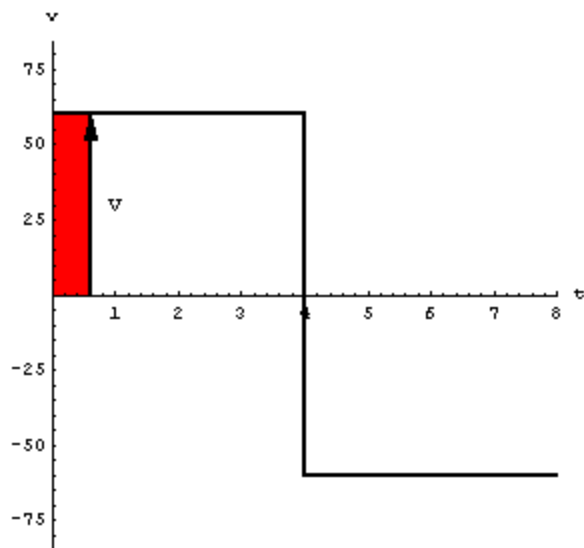
```
v[t_] = Which[t ≤ 4, 60, t > 4, -60];
```

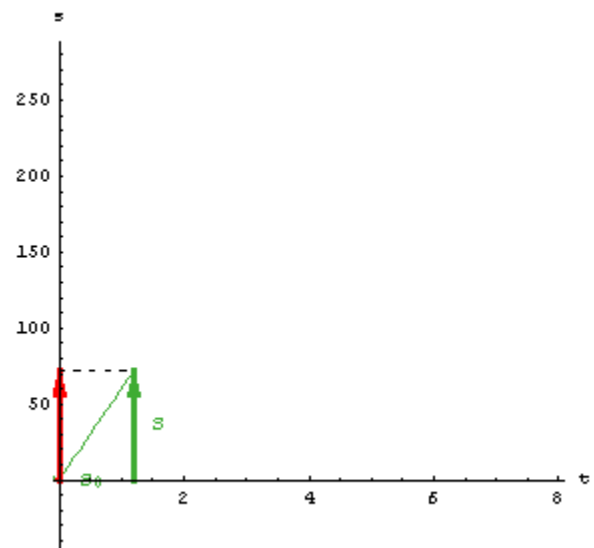
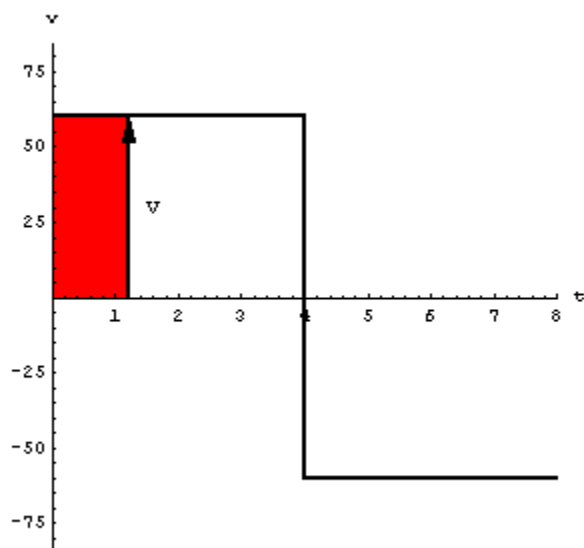
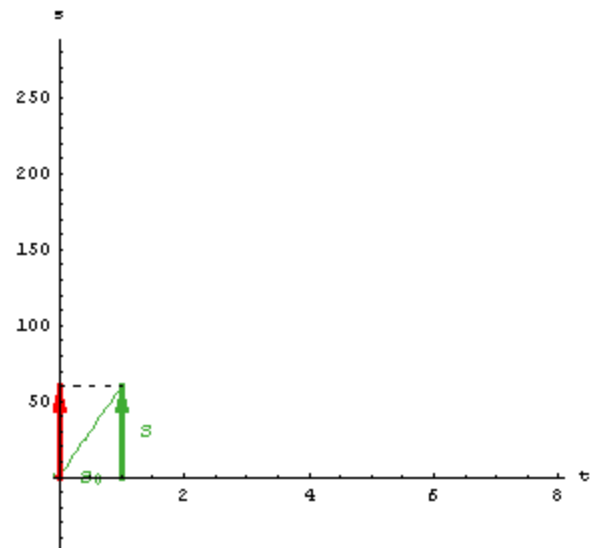
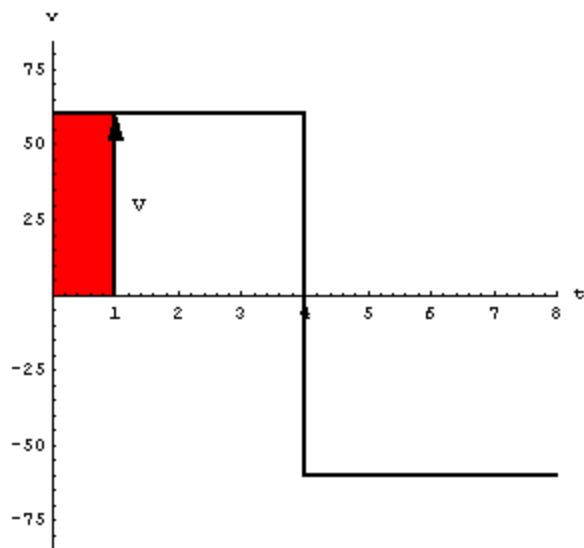
```
s0 = 0;
```

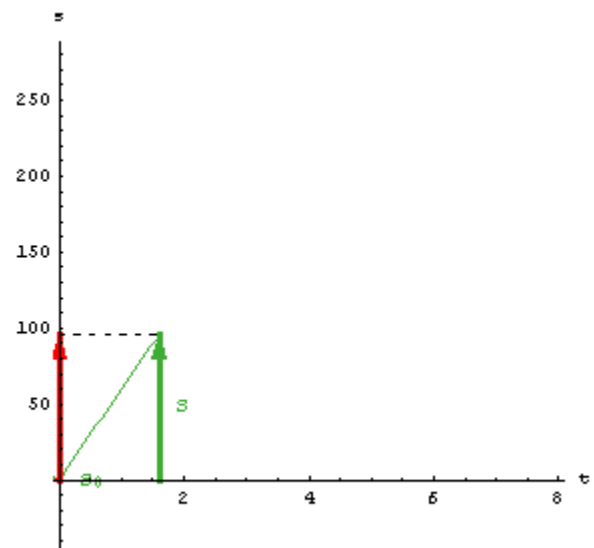
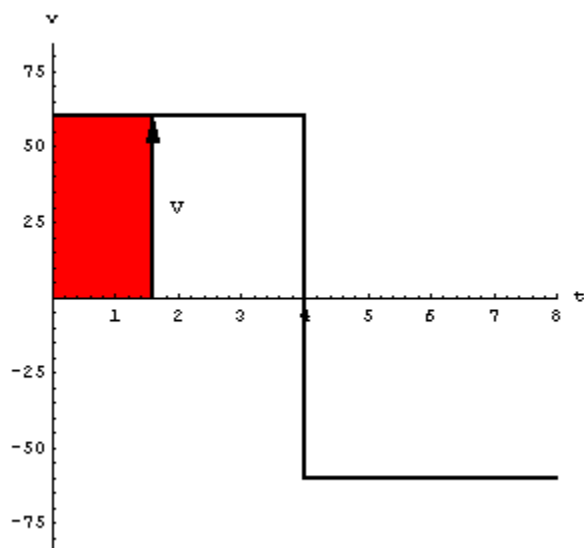
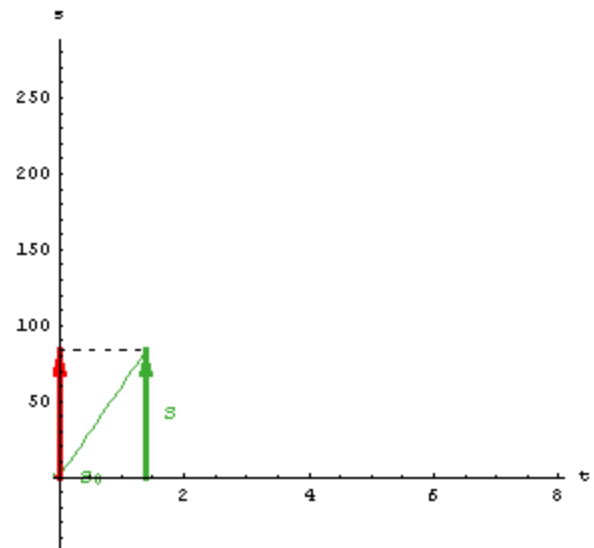
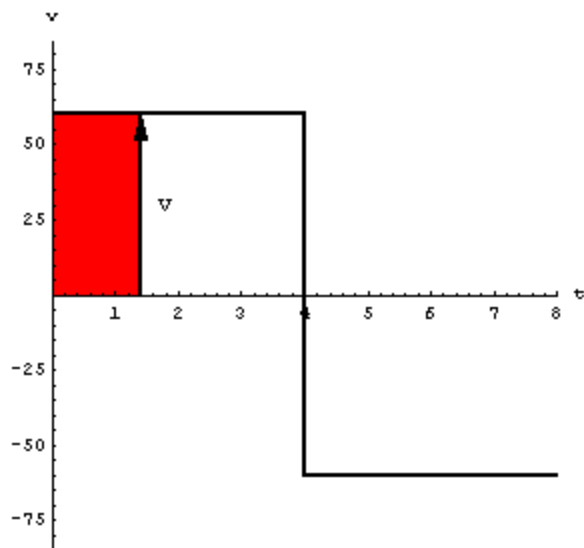
```
position[v[t], {t, 0, 8}, s0];
```

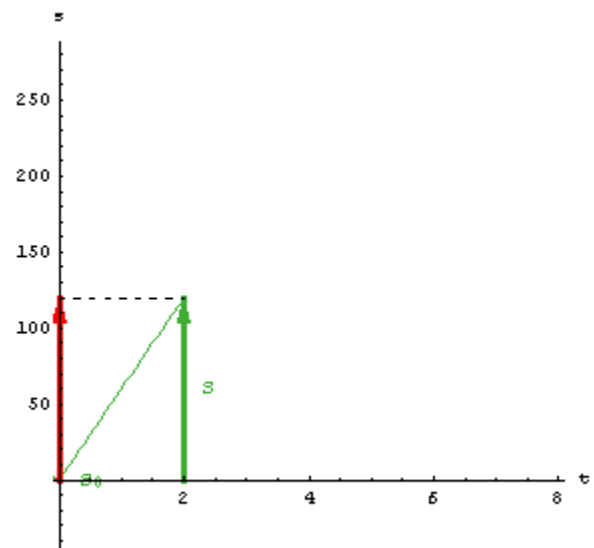
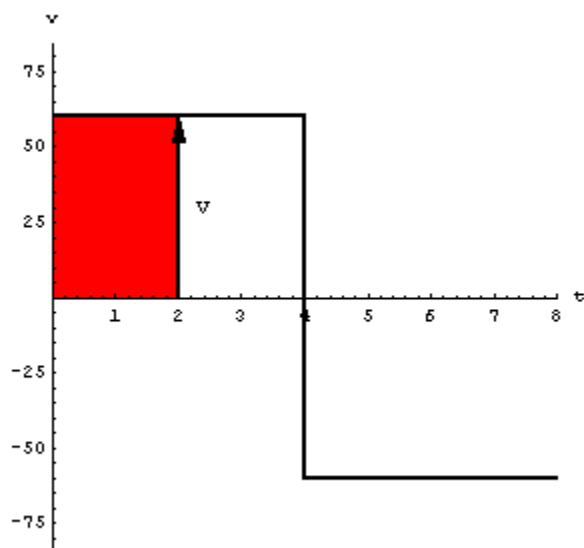
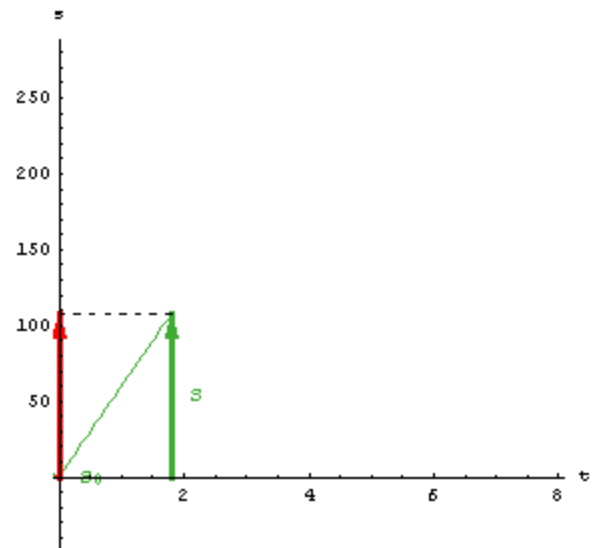
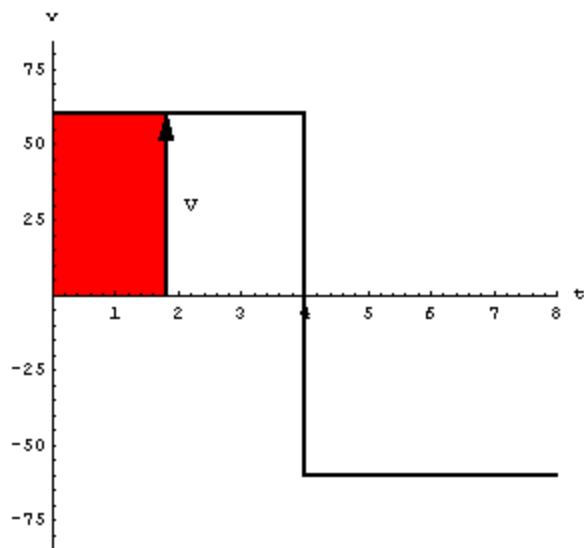


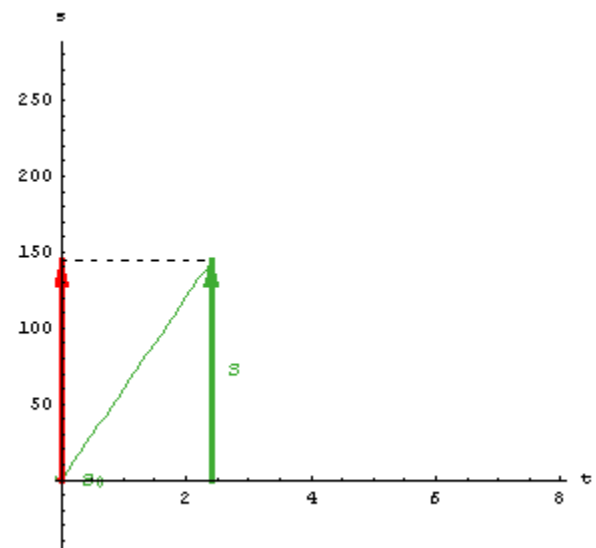
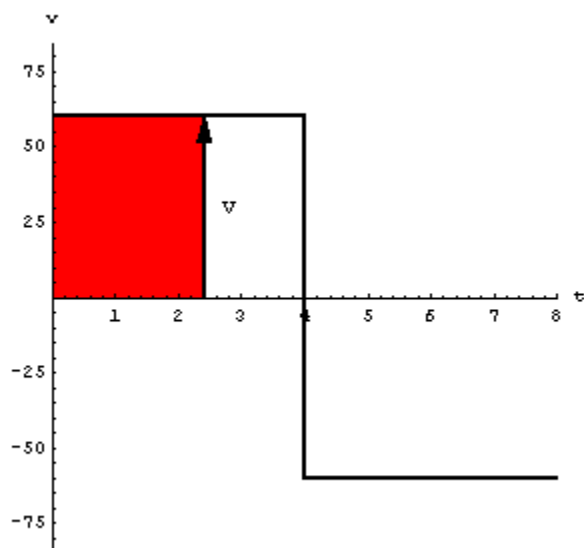
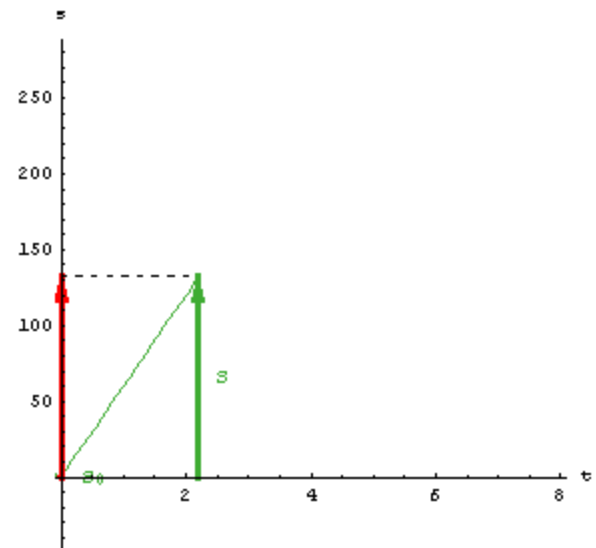
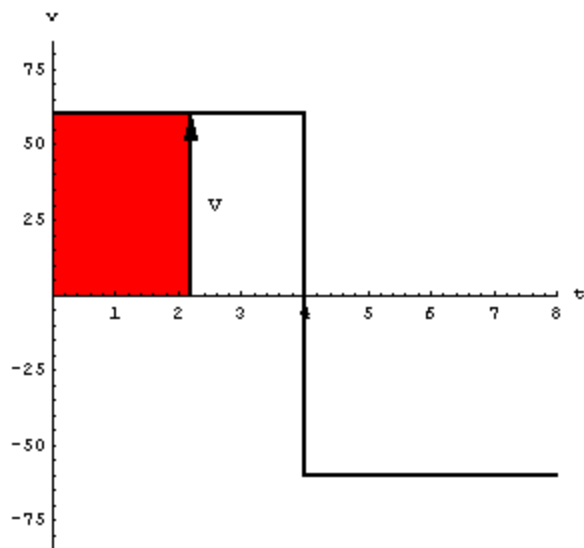


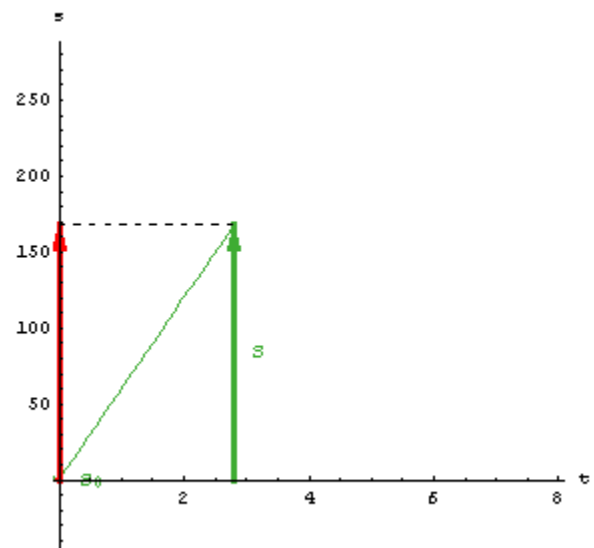
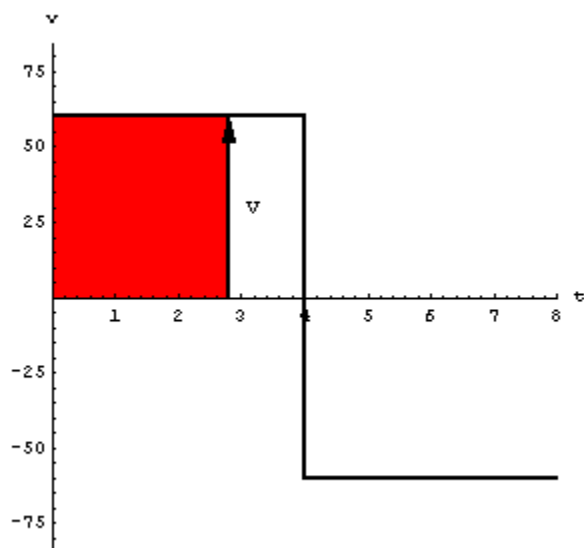
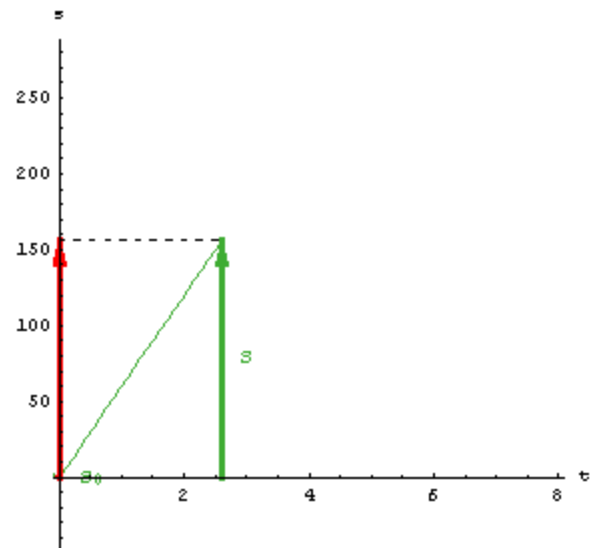
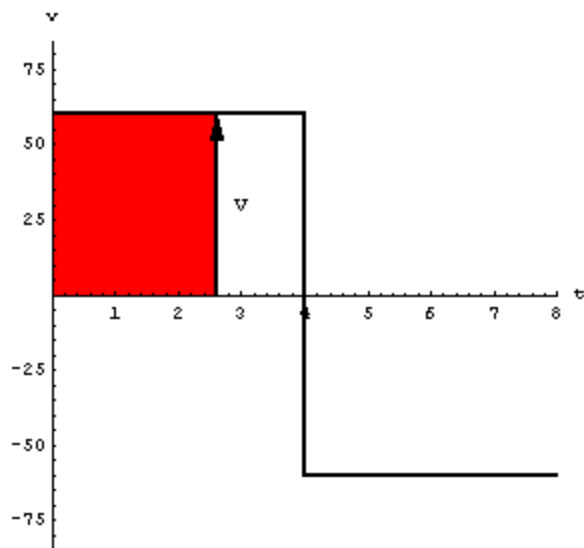


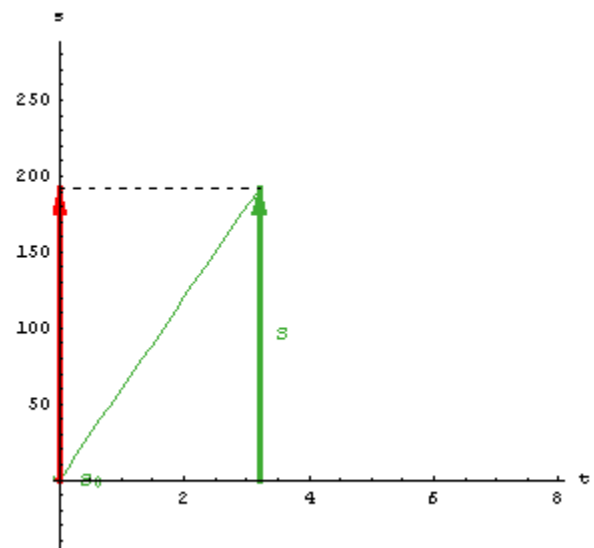
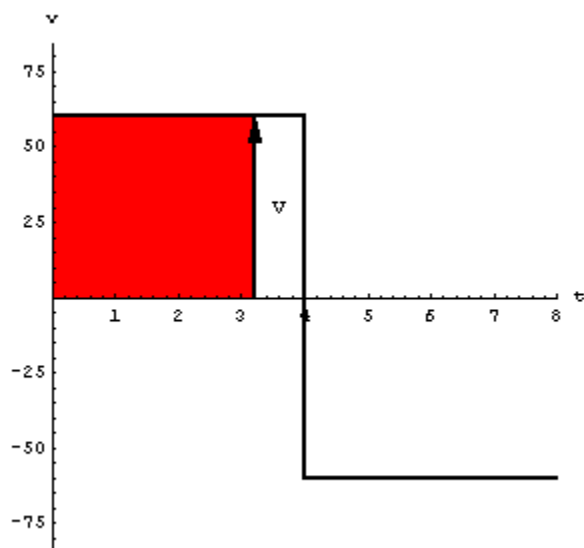
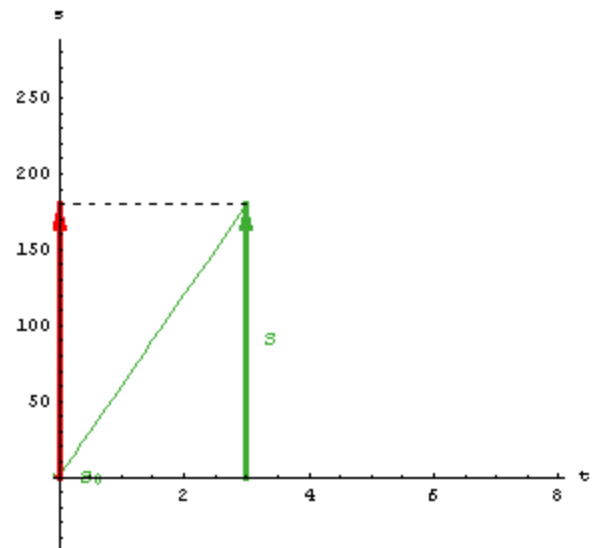
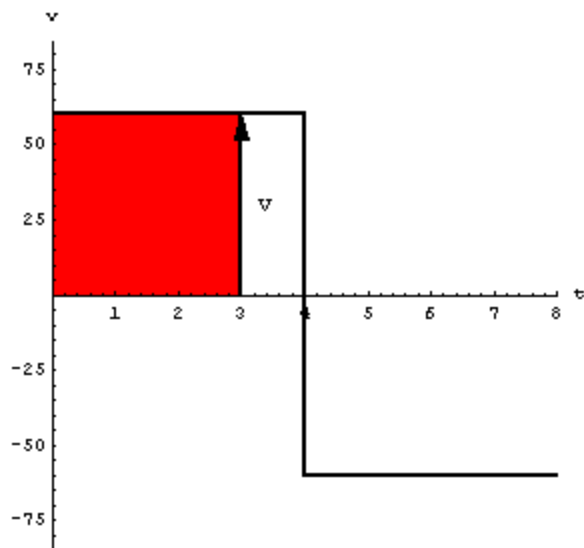


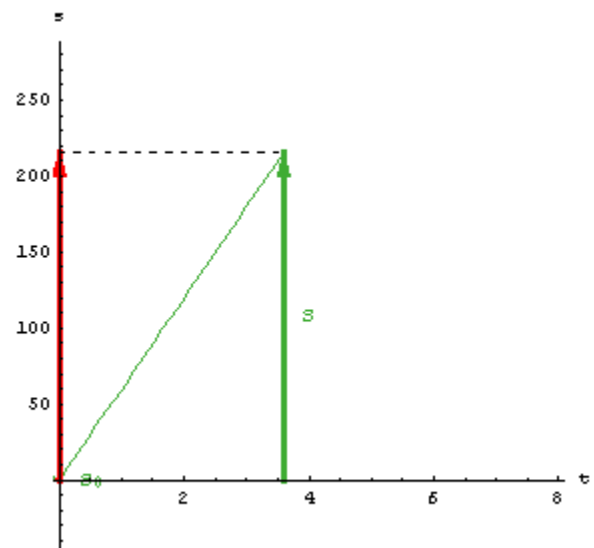
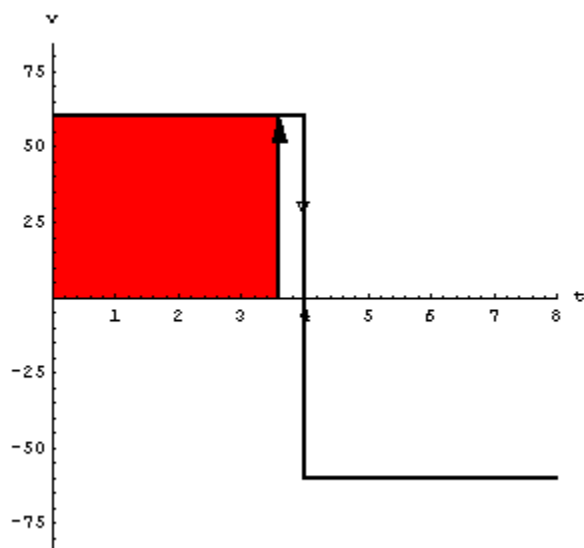
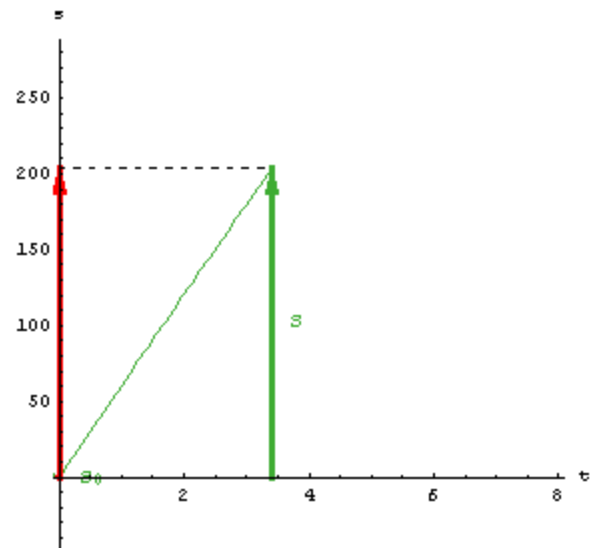
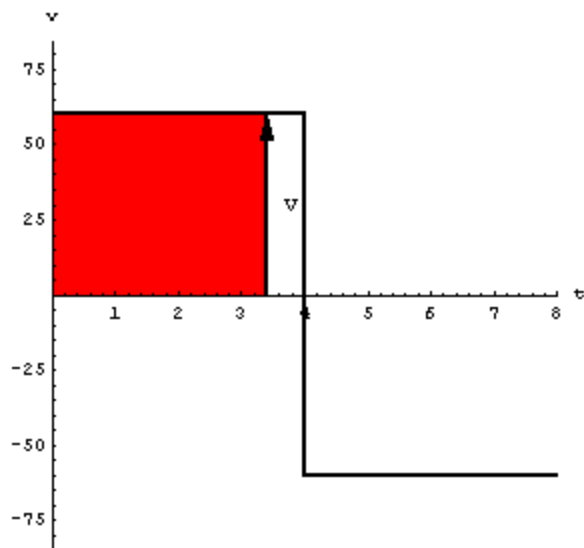


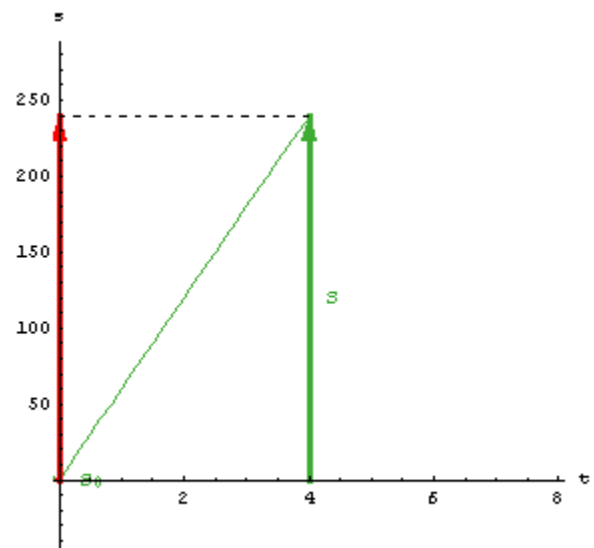
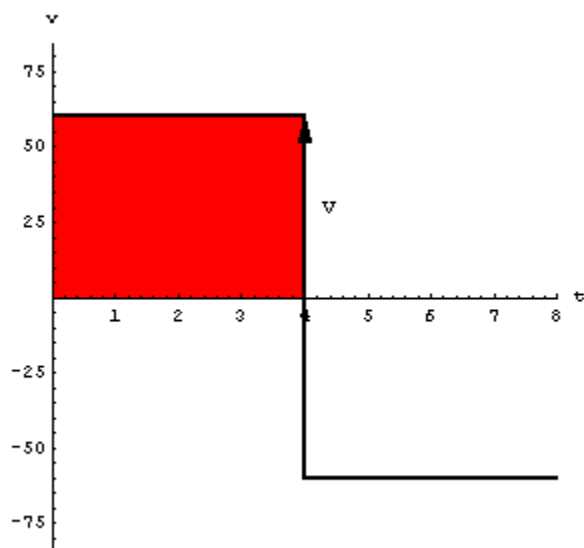
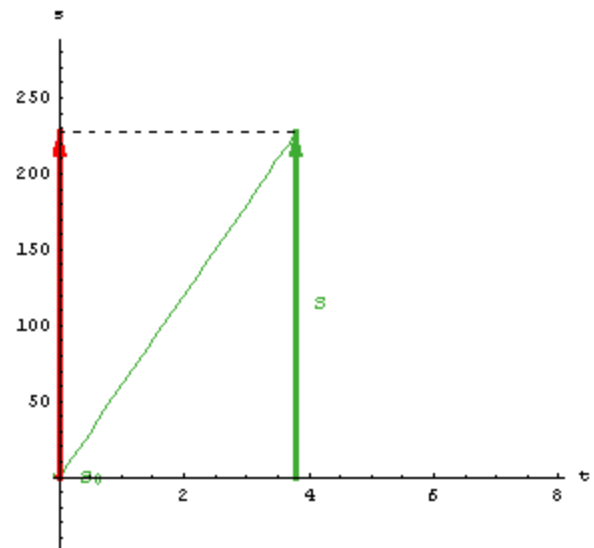
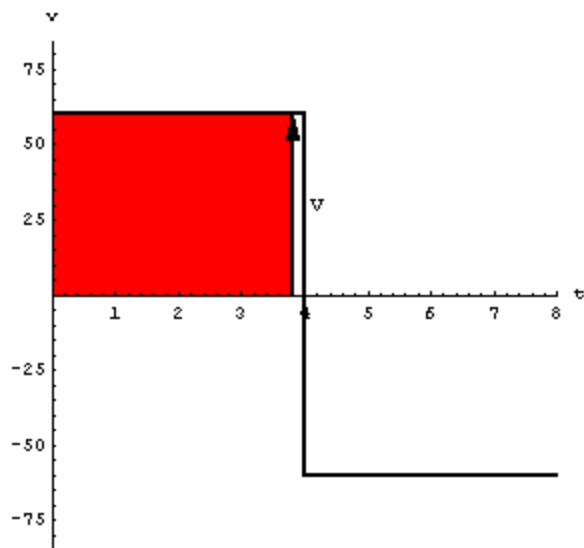


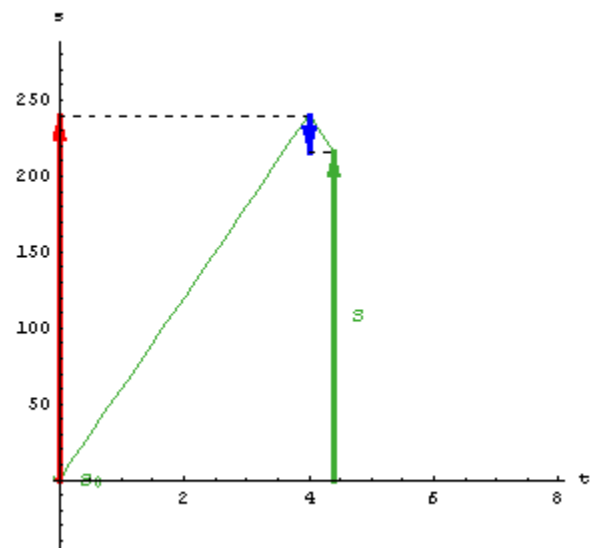
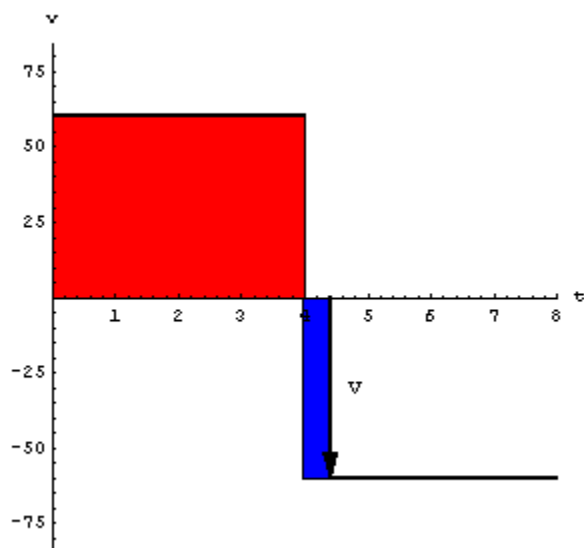
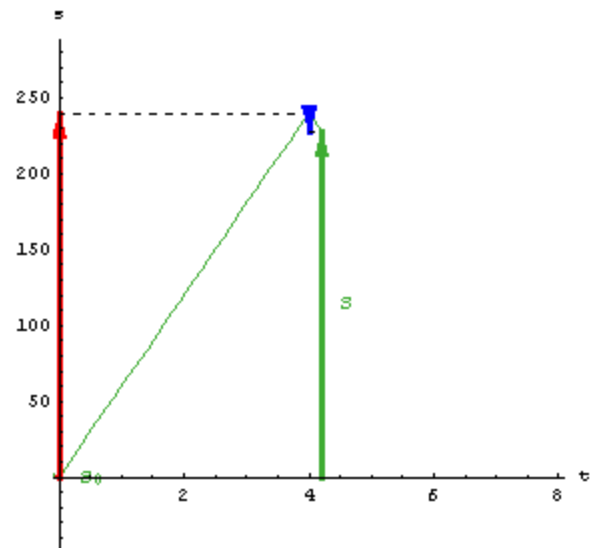
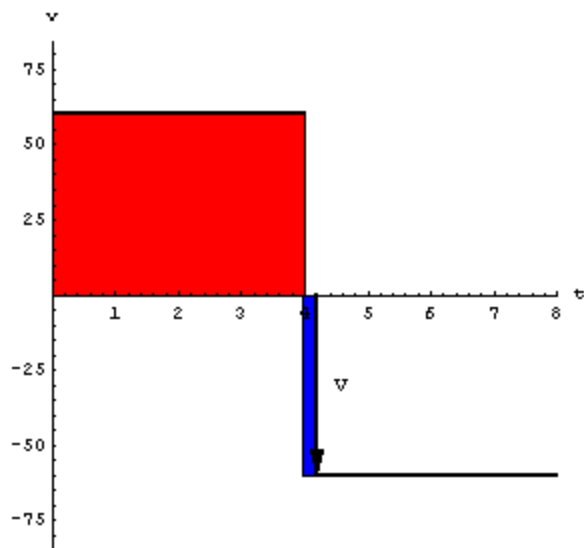


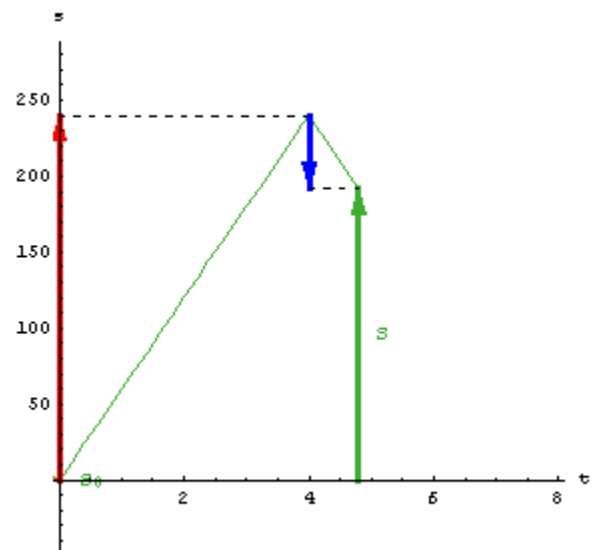
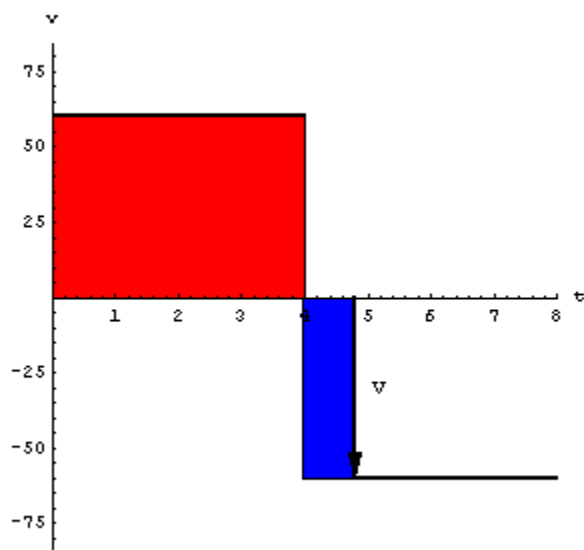
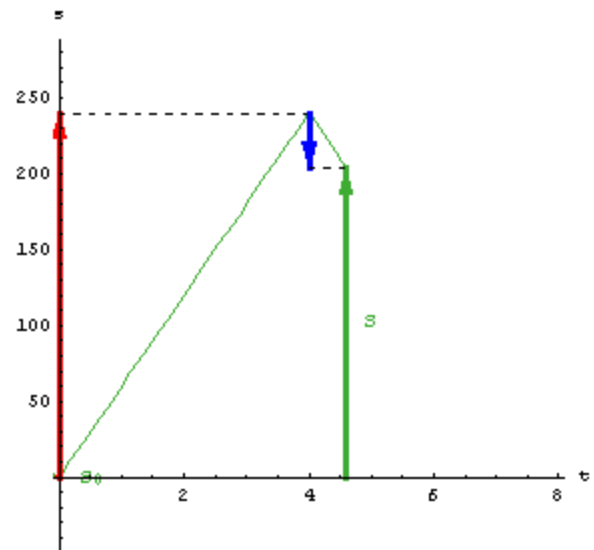
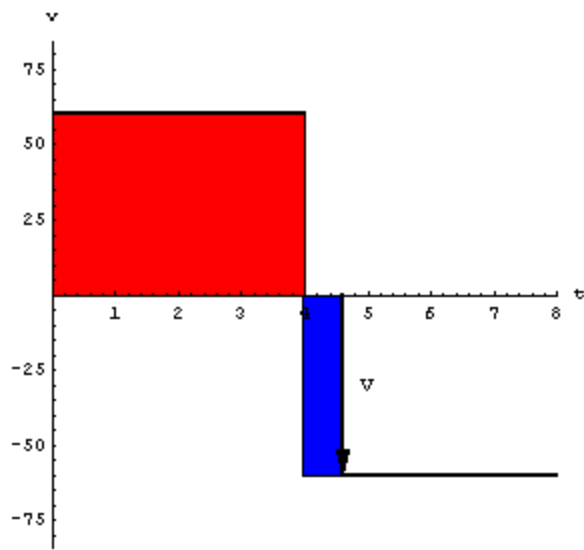


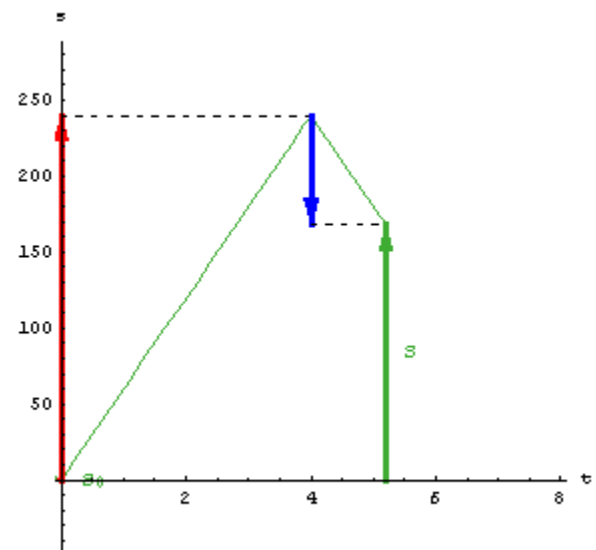
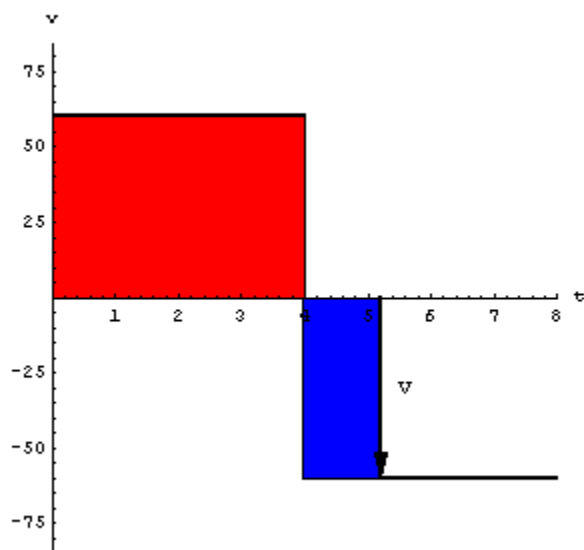
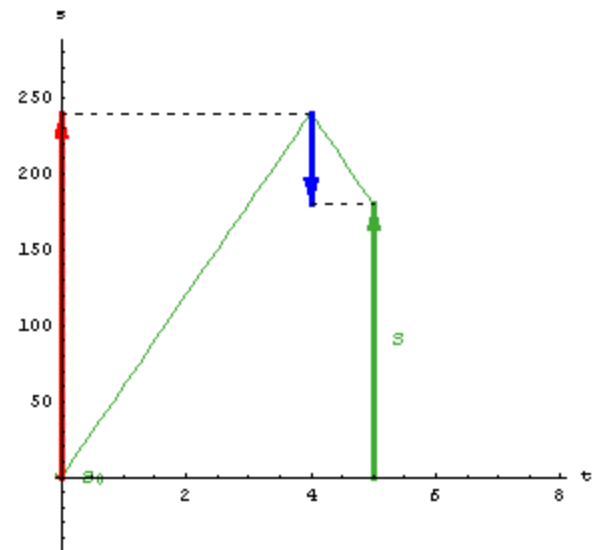
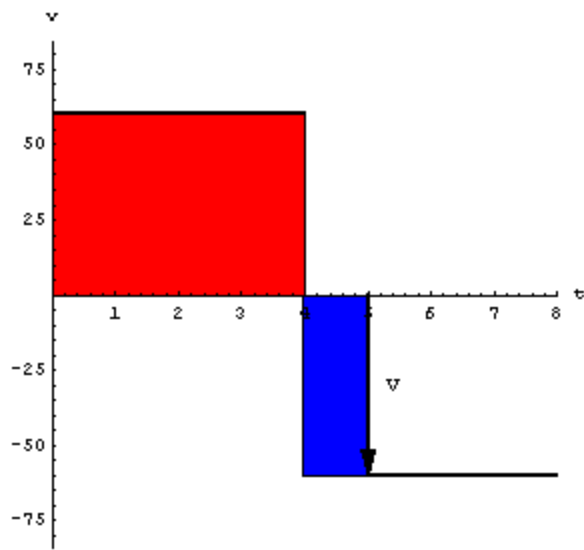


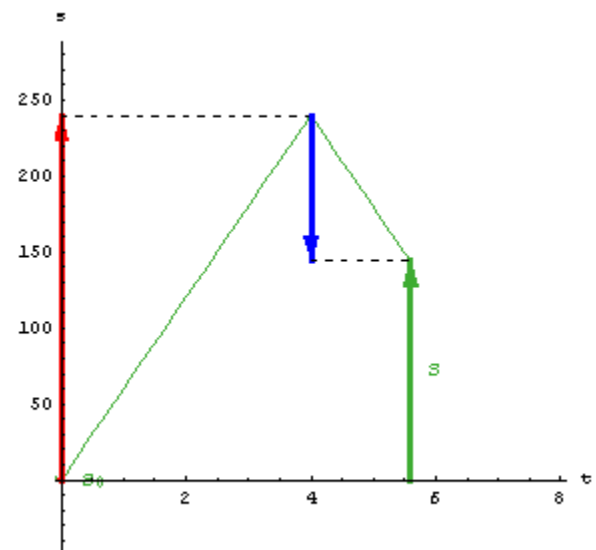
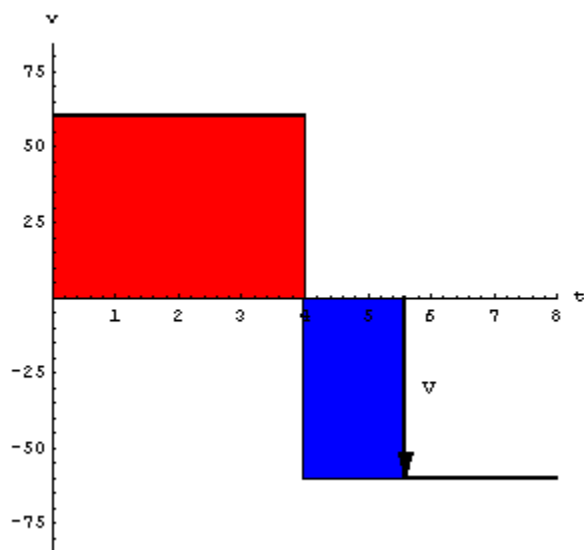
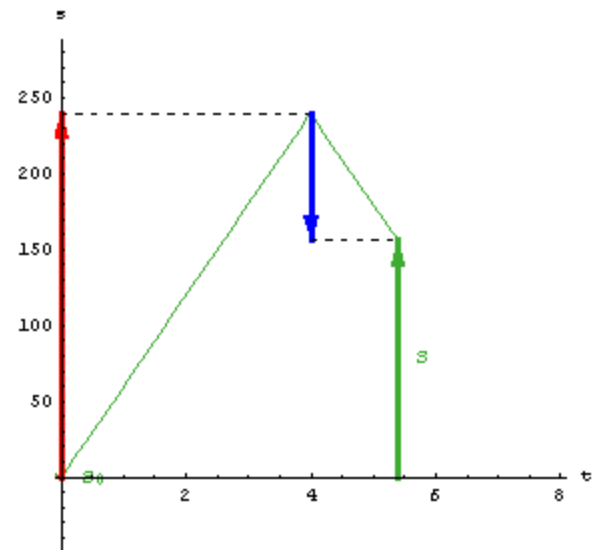
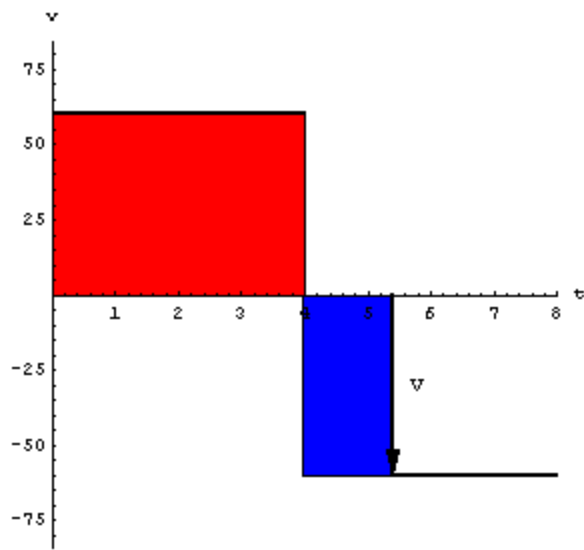


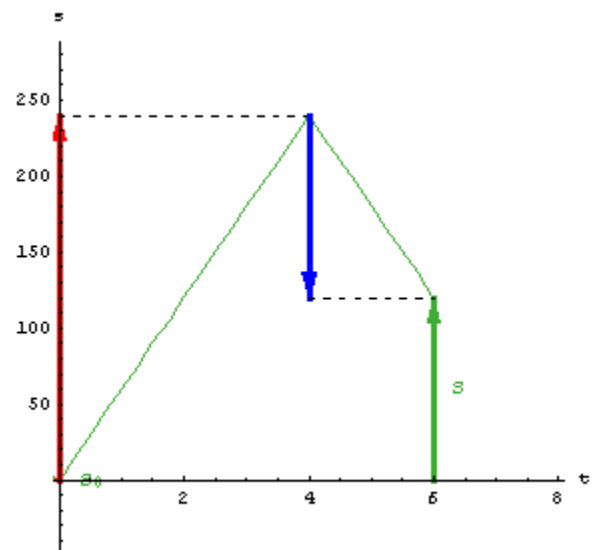
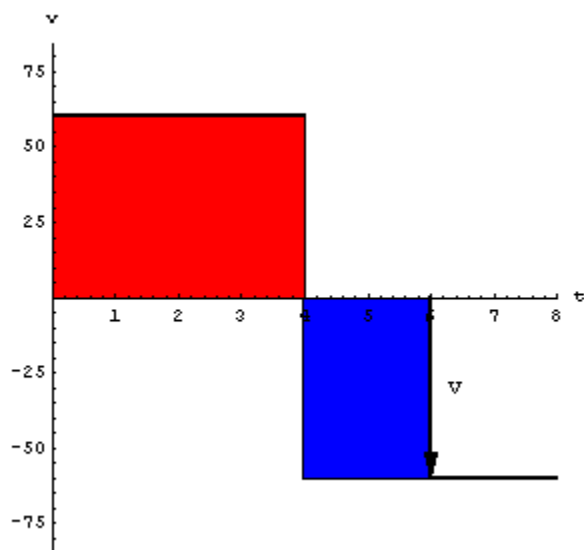
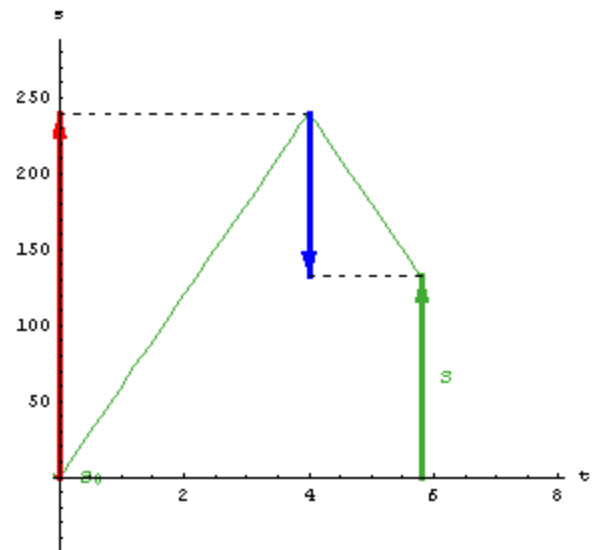
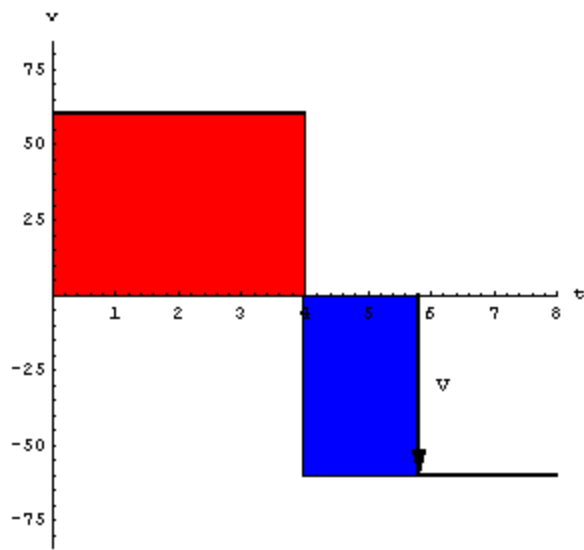


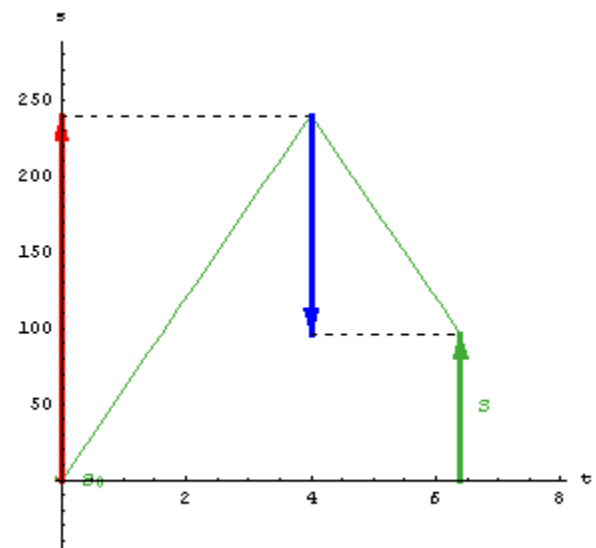
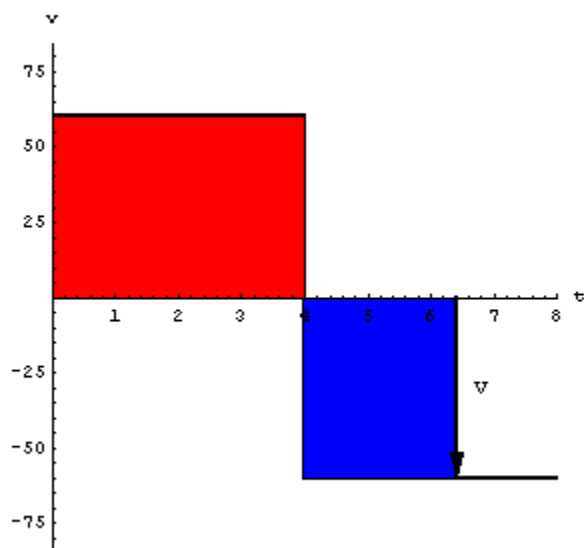
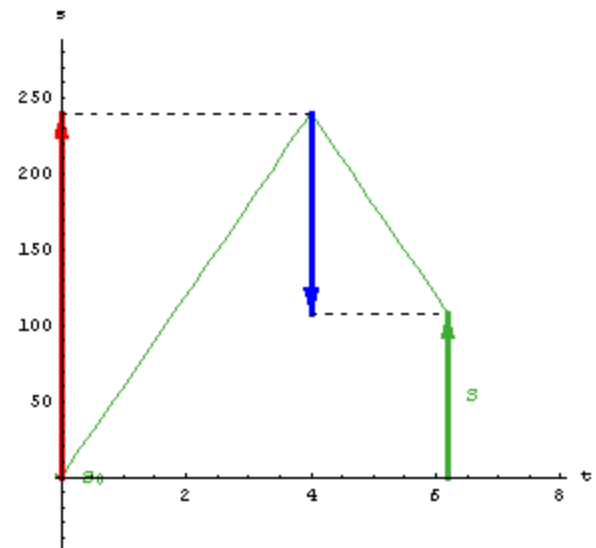
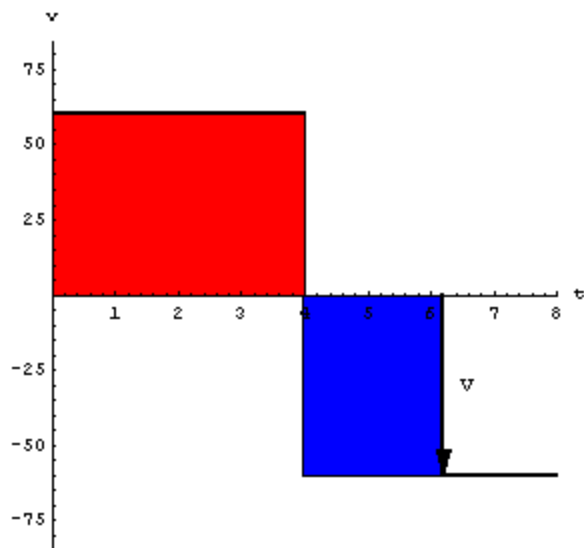


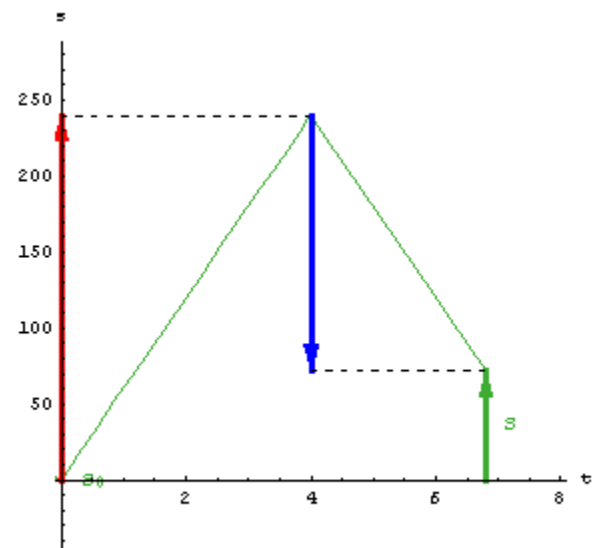
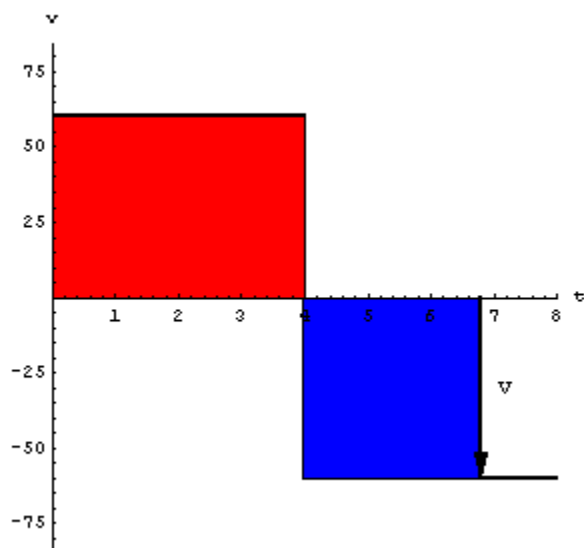
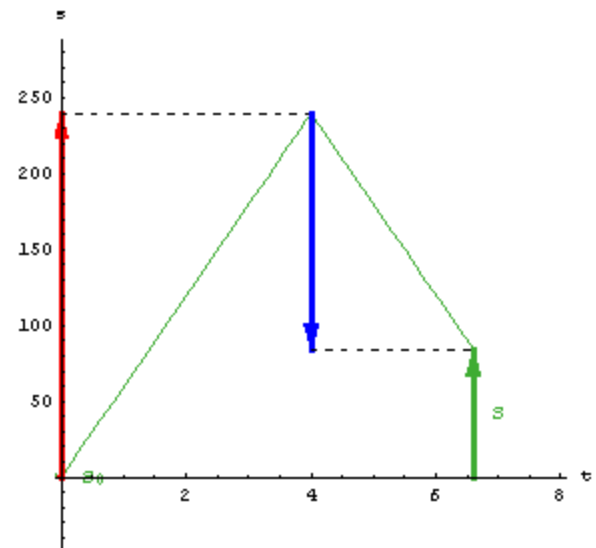
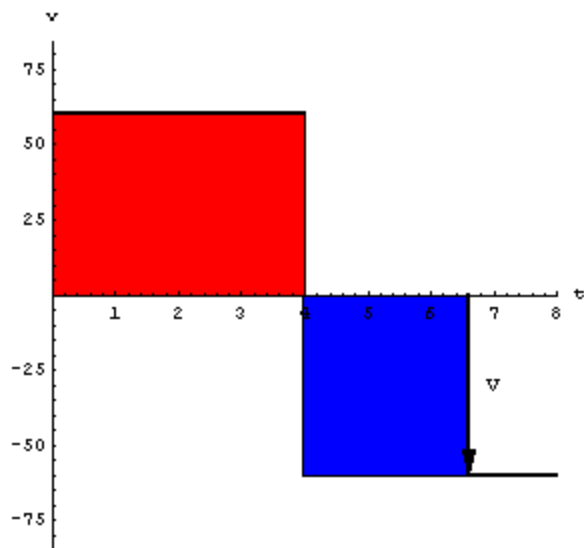


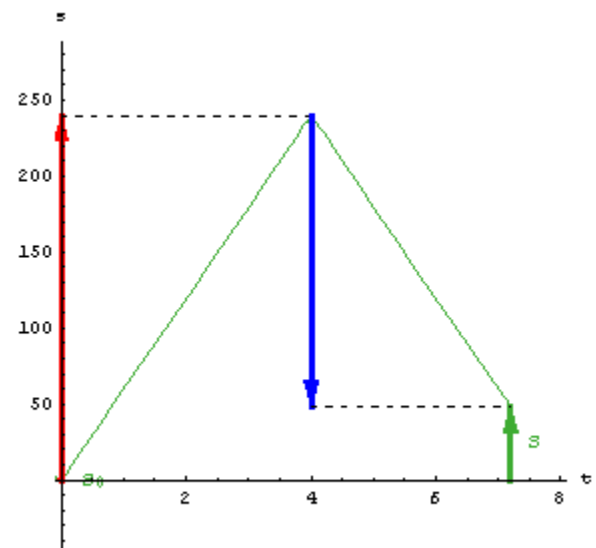
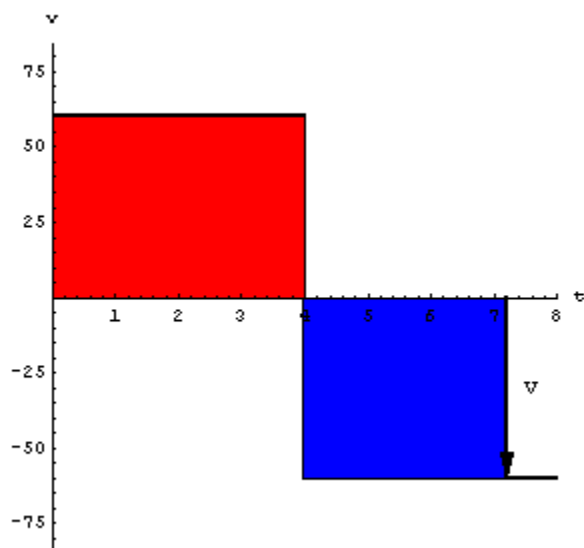
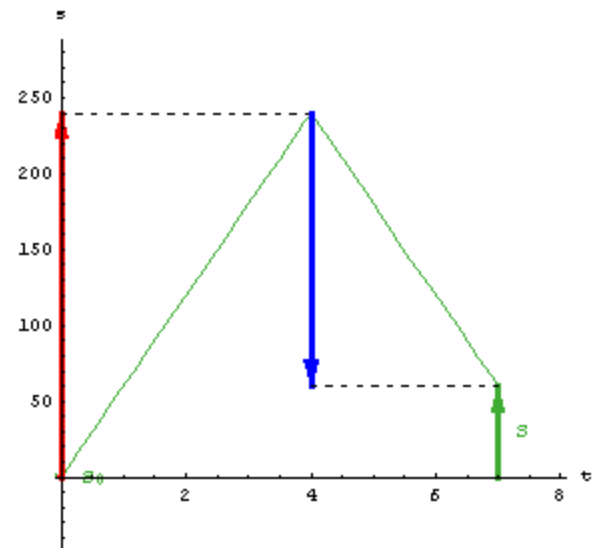
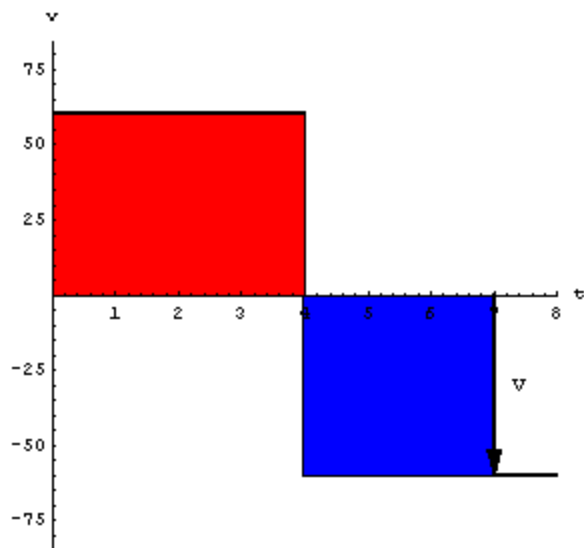


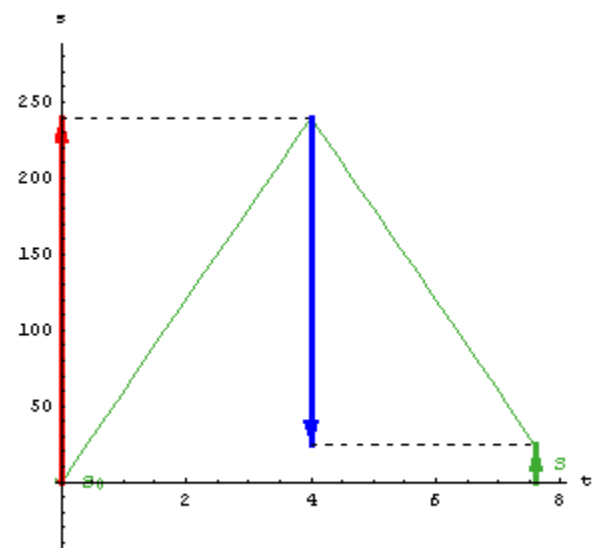
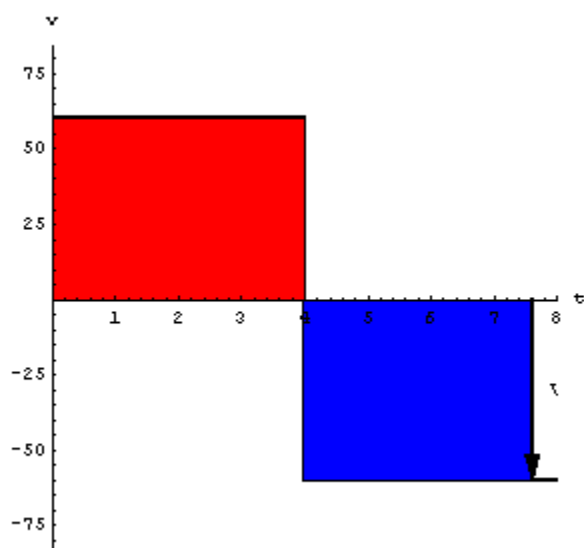
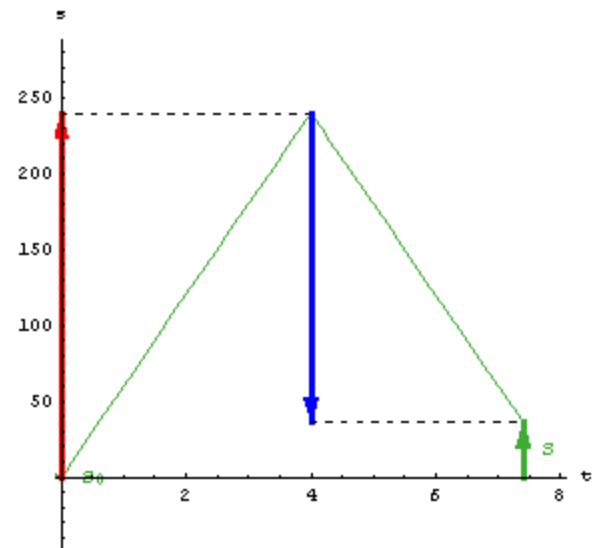
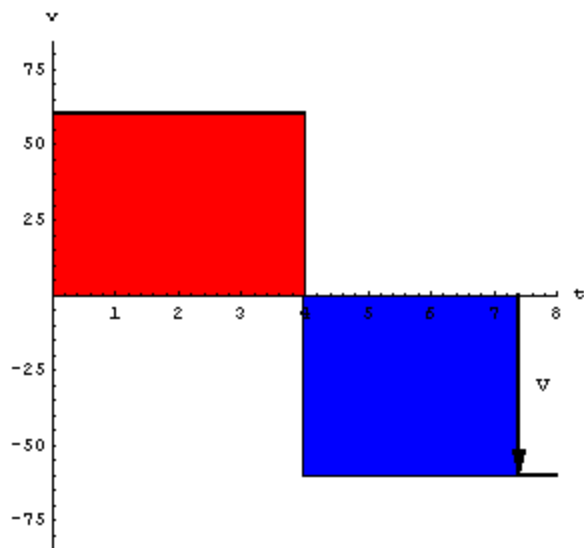


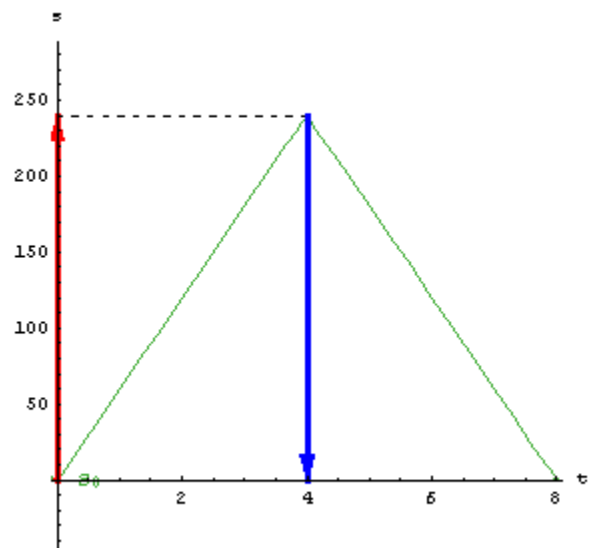
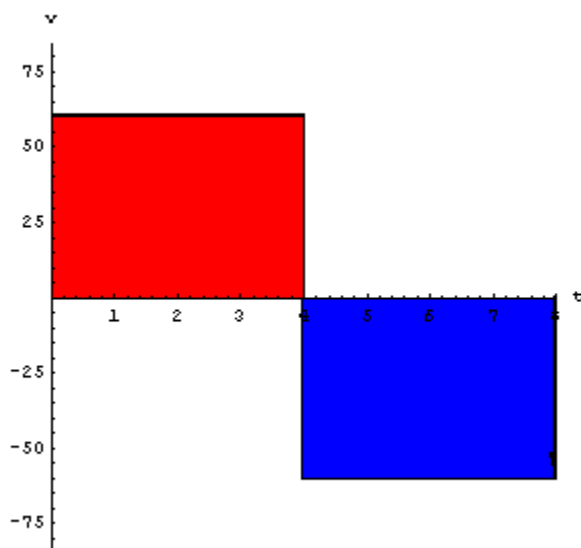
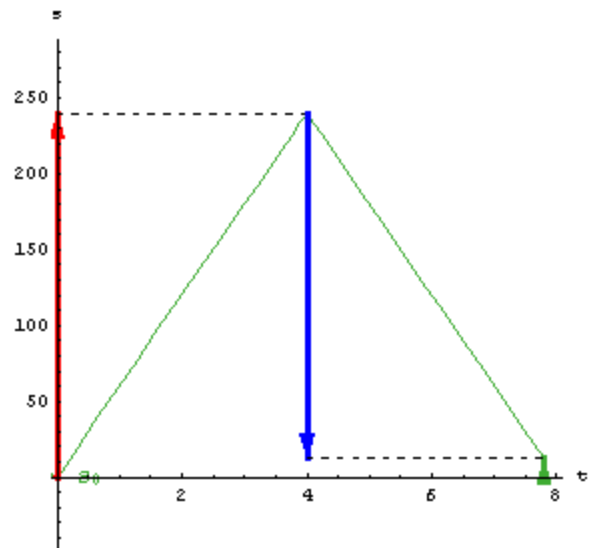
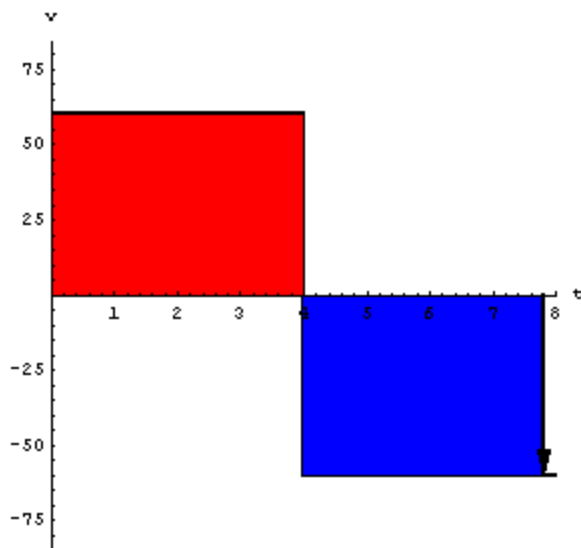












Part II: Constant Acceleration

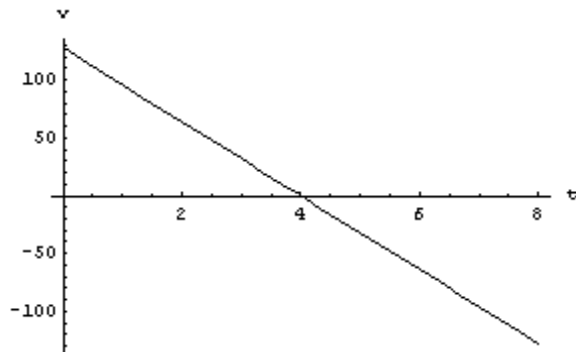
Imagine throwing an object straight up from the ground with an initial speed of 128 ft/sec. You can give its velocity as the following function of time.

`In[30]:=`

`Clear[v];`

```
v[t_] = 128 - 32 * t;
```

```
Plot[v[t], {t, 0, 8}, AxesLabel → {"t", "v"}];
```



Assuming that the object is thrown straight up from the ground, which is taken as the reference point (i.e., $s = 0$), you can determine the position function and graph it.

```
In[34]:=
```

```
Clear[s];
```

```
s0 = 0;
```

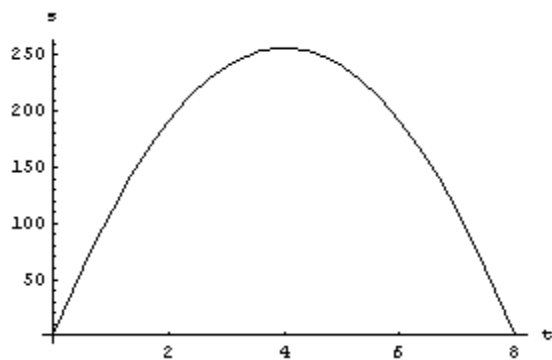
```
s[t_] = Integrate[v[u], {u, 0, t}] + s0
```

```
Out[36]=
```

```
128 t - 16 t2
```

```
In[37]:=
```

```
Plot[s[t], {t, 0, 8}, AxesLabel → {"t", "s"}];
```

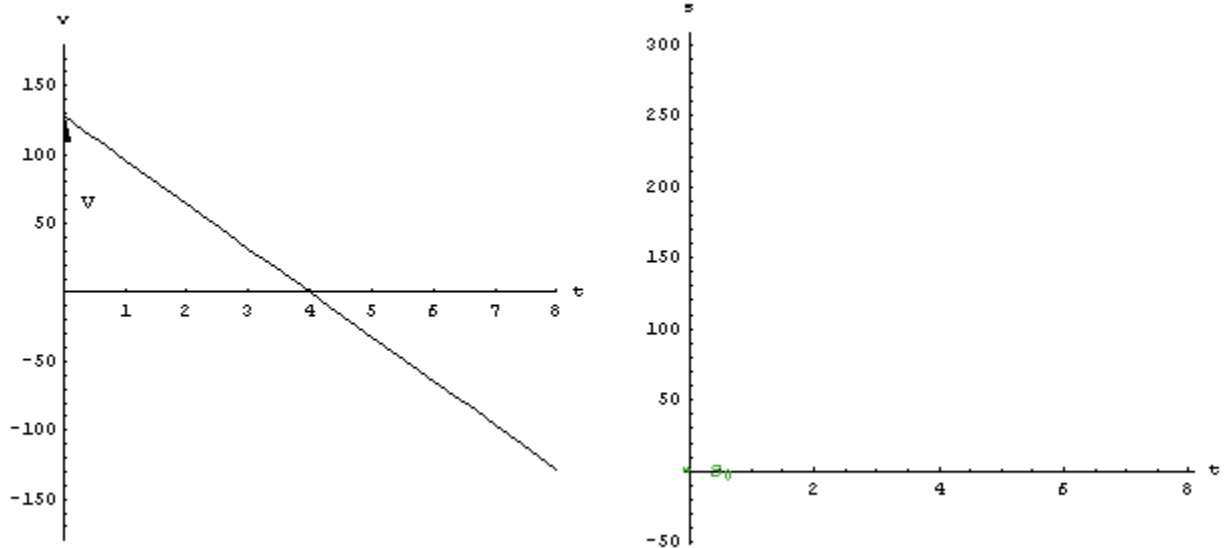


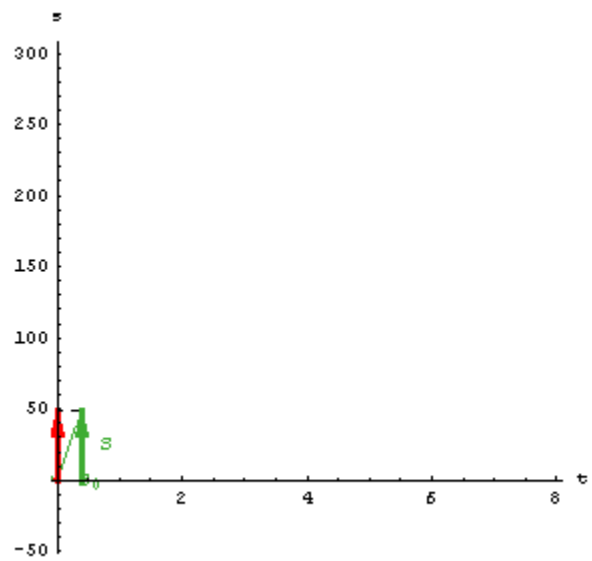
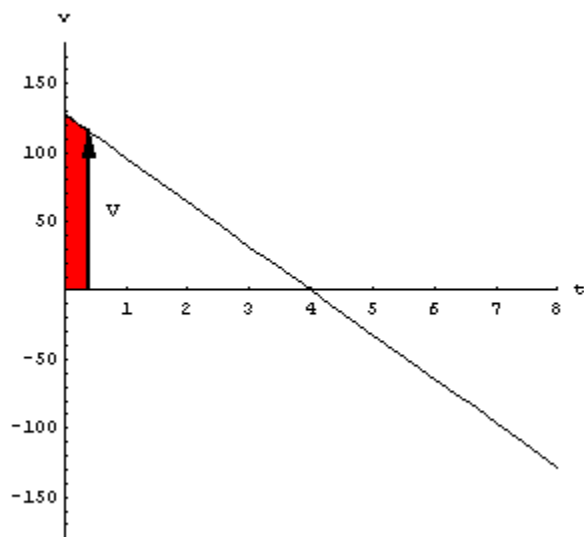
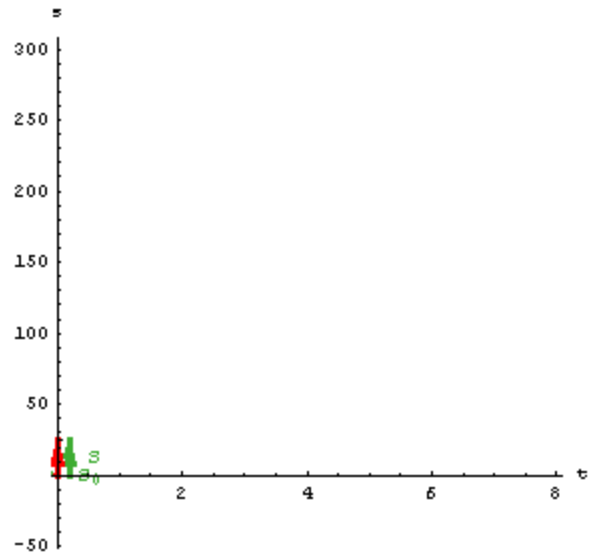
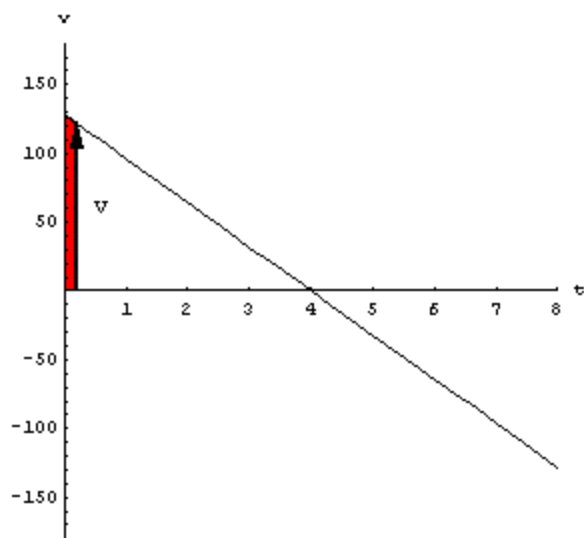
Now use **position[]** to illustrate the accumulation of displacements on the graphs of the velocity and the position functions. On the velocity graph, shaded areas represent displacements or changes in position of the moving object. Red areas are numerically equal to positive displacements of the moving object, and blue areas are numerically equal to negative displacements. On the position graph, the red and blue colored arrows represent the displacements of the object. Red arrows show positive displacements and blue arrows show negative displacements. The red shaded areas on the velocity graph are numerically equal to the lengths of the corresponding red arrows on the position graph. Similarly, the blue areas are numerically equal to the lengths of the corresponding blue arrows.

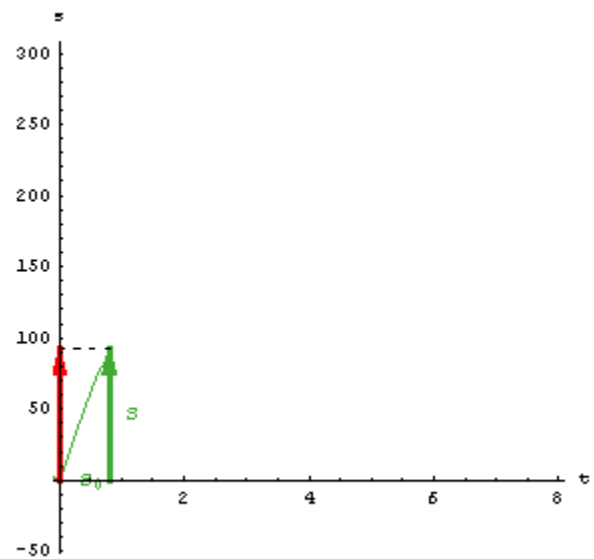
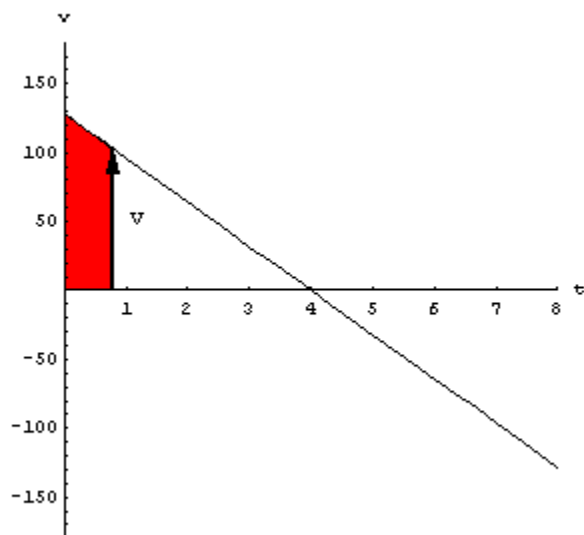
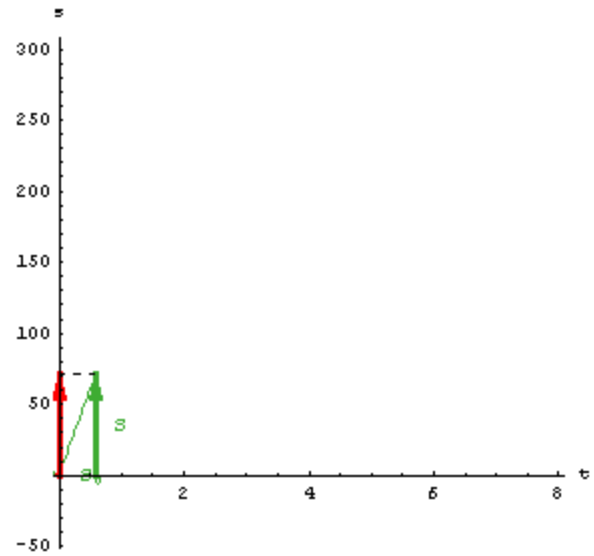
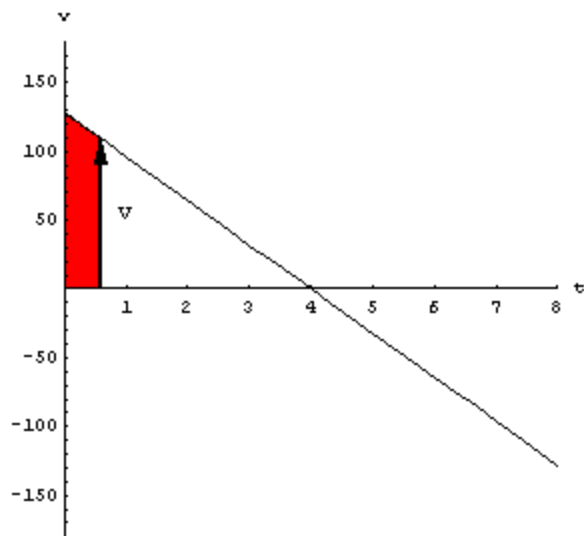
Note: The next command generates a lot of graphics, which may fill up your computer's memory. Before executing the next command, pull down the Kernel menu and select Delete All Output. All computed values are saved when you do this.

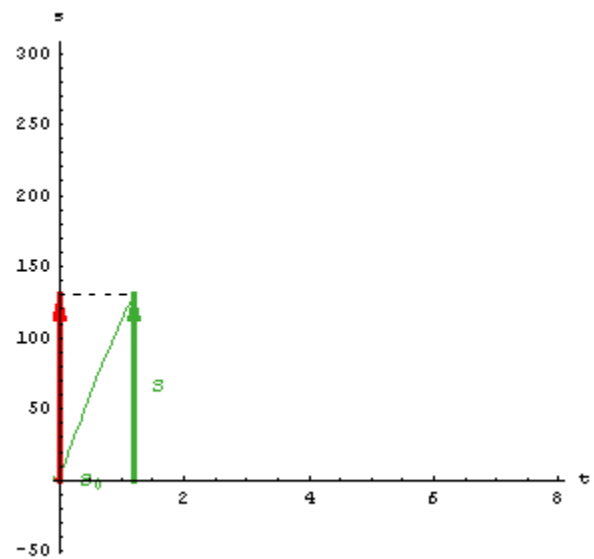
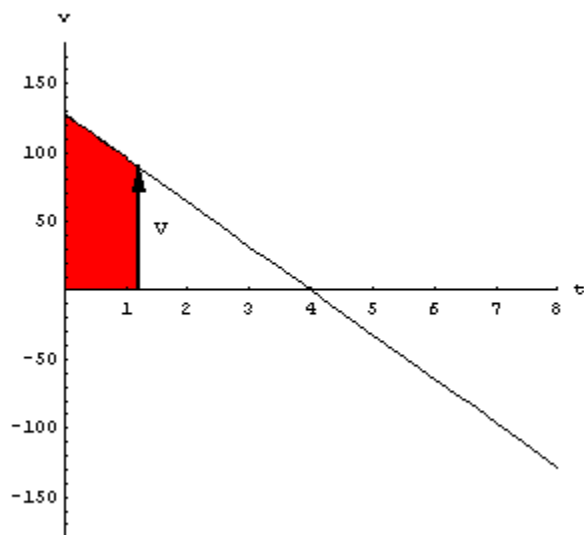
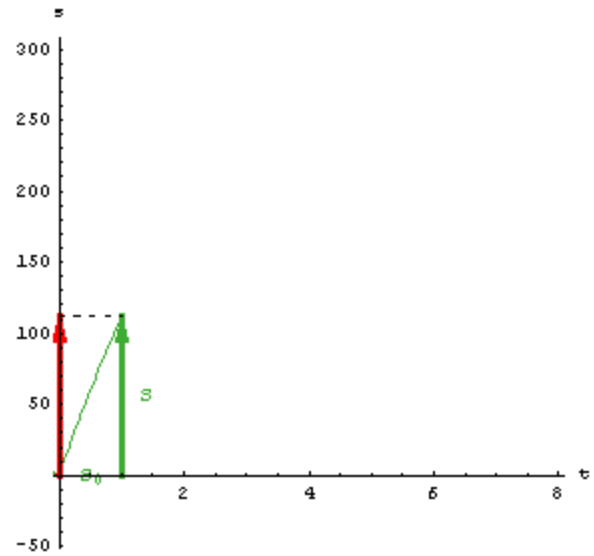
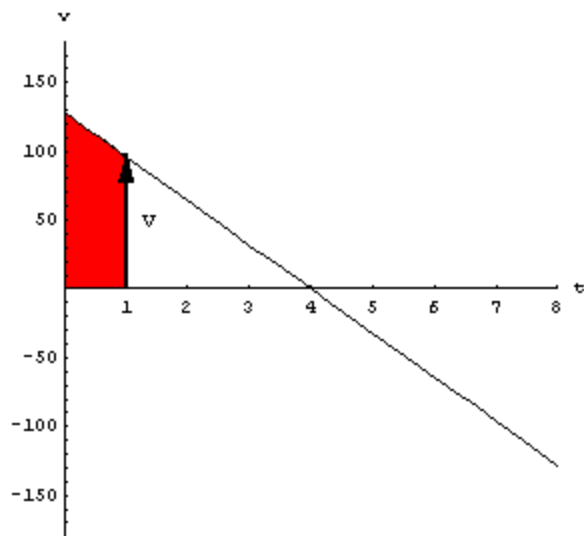
In[38]:=

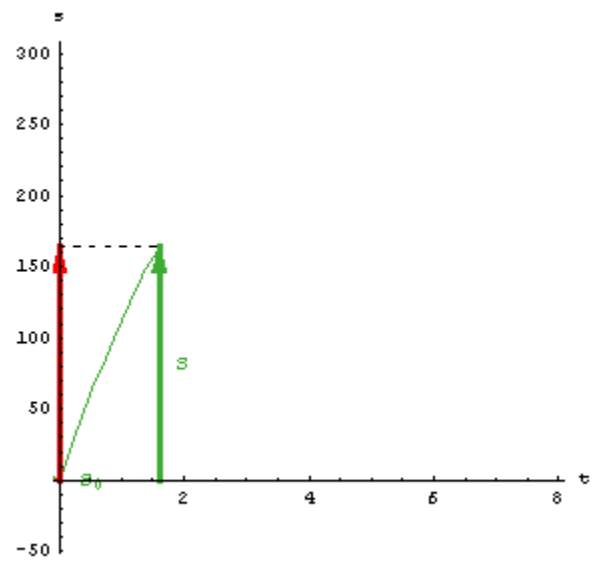
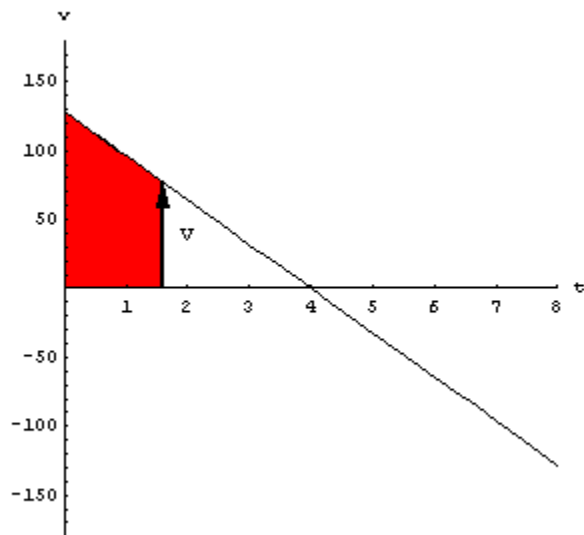
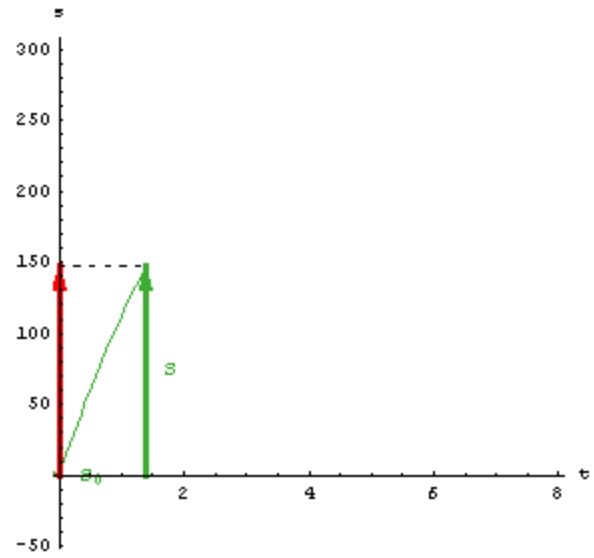
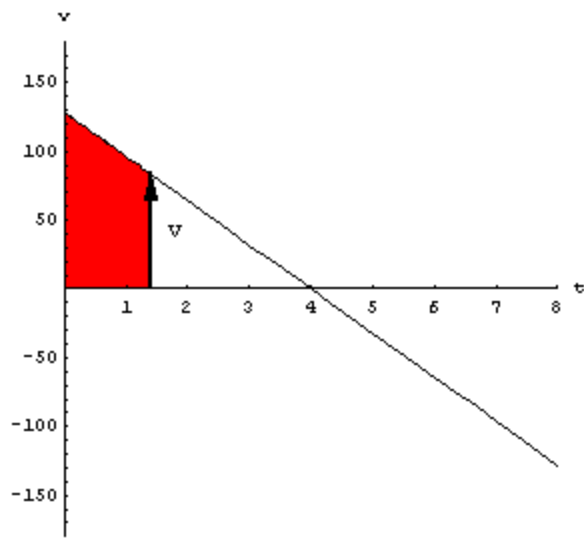
```
position[v[t], {t, 0, 8}, s0];
```

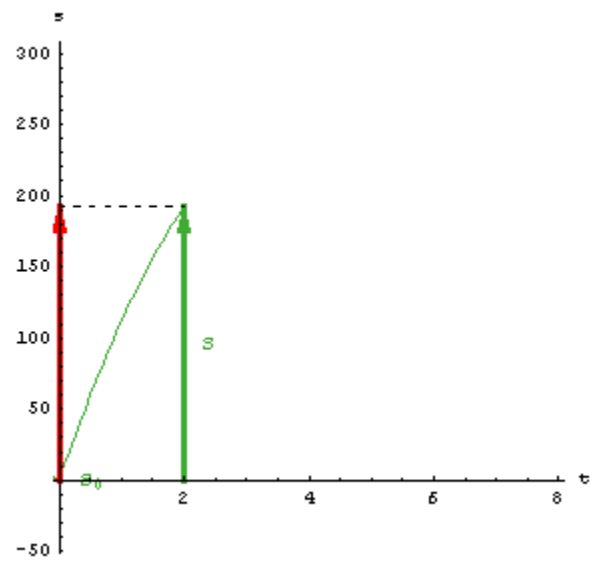
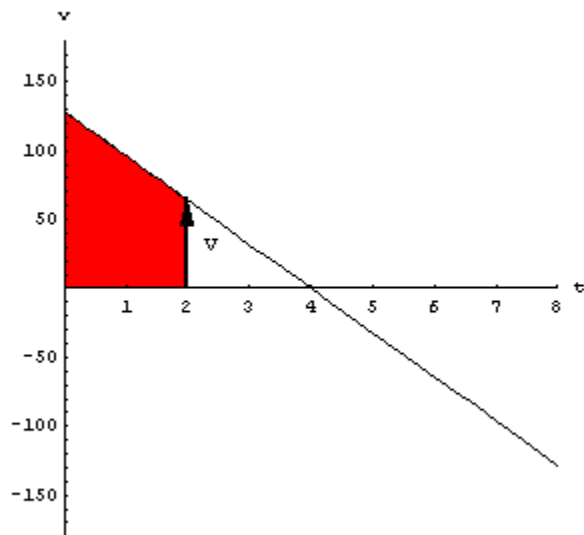
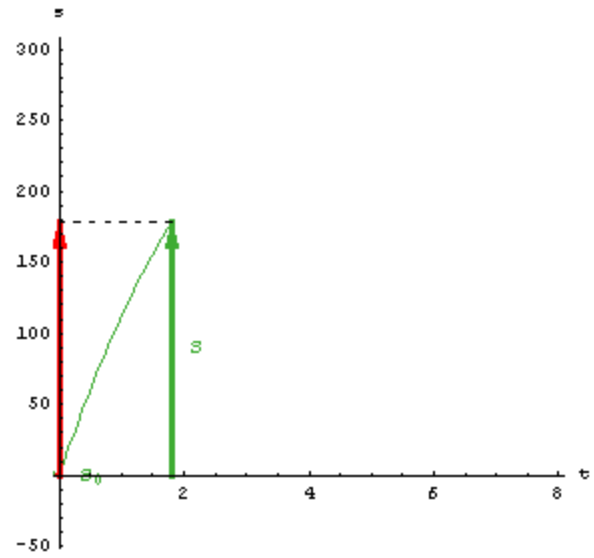
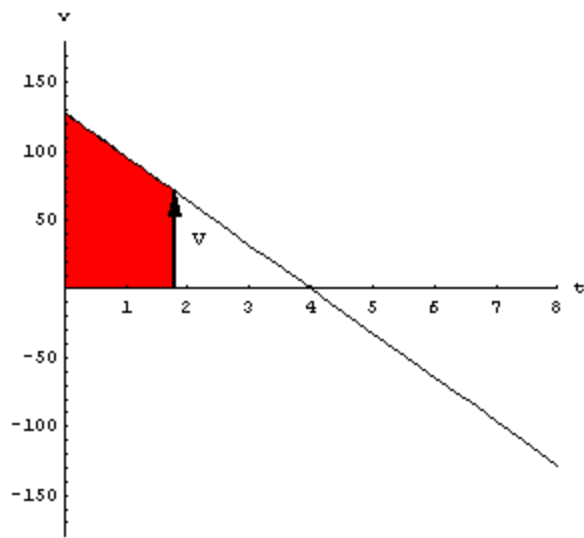


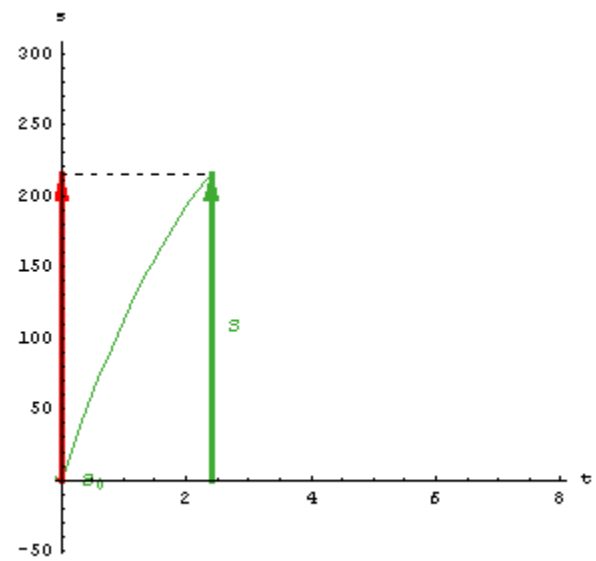
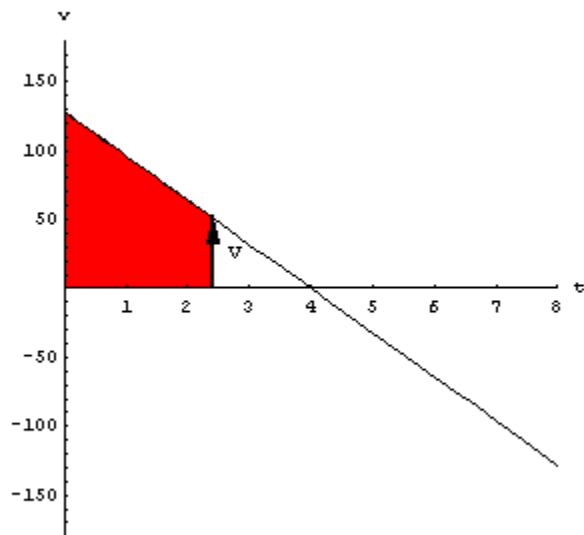
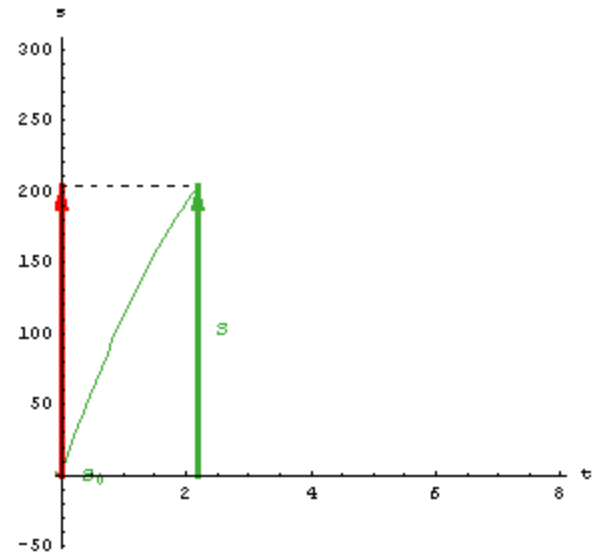
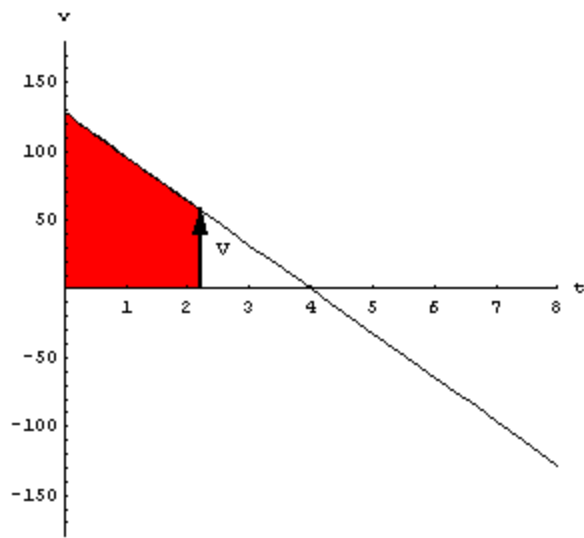


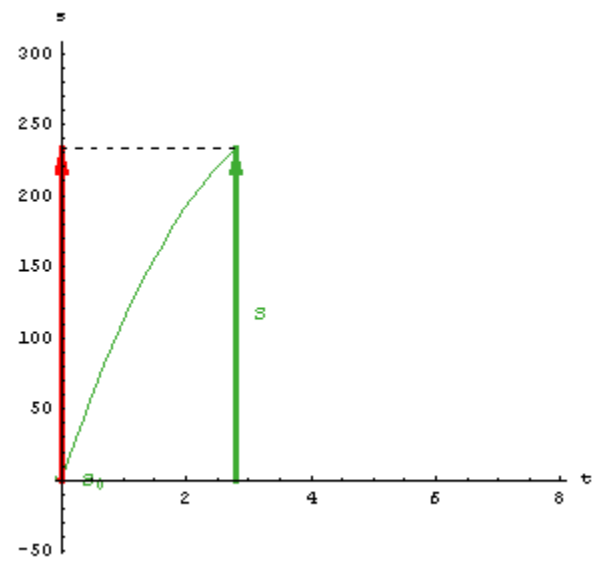
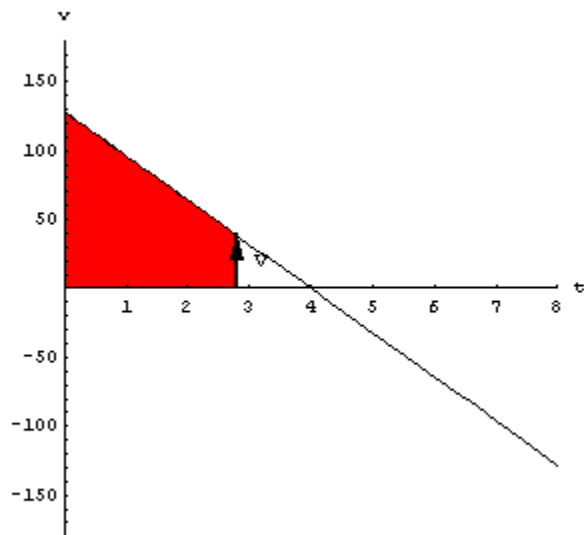
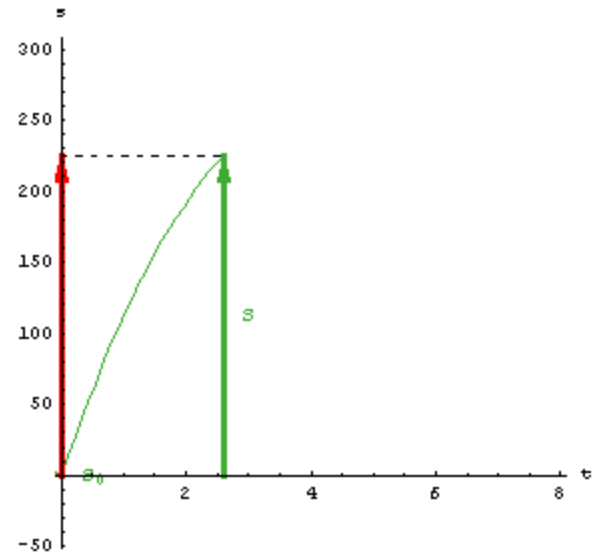
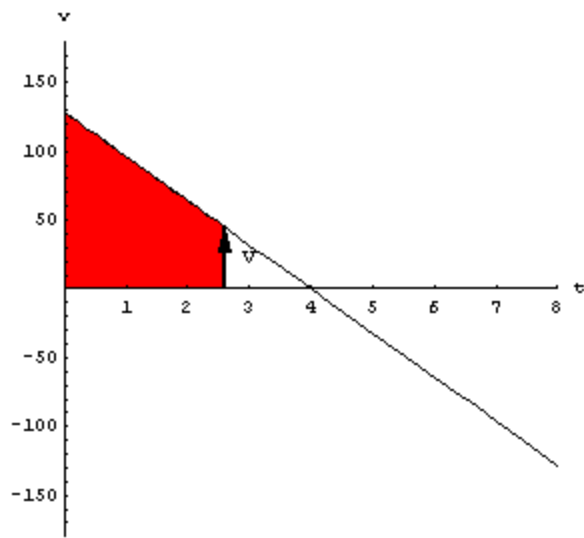


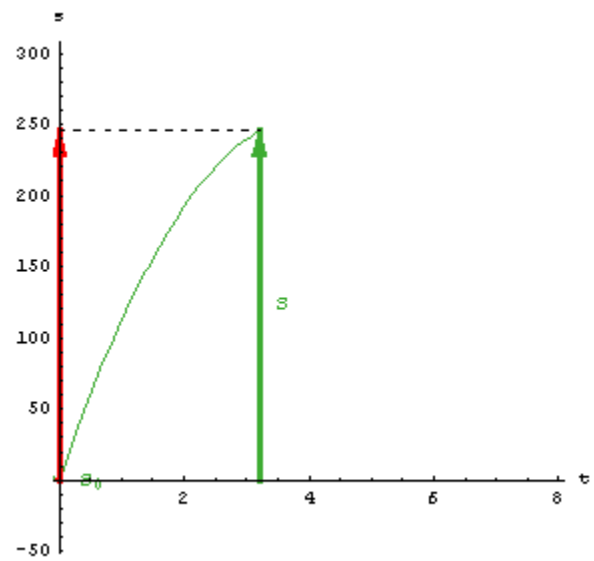
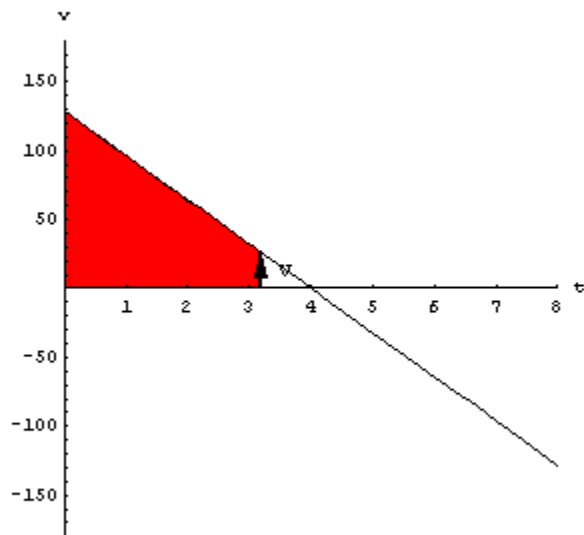
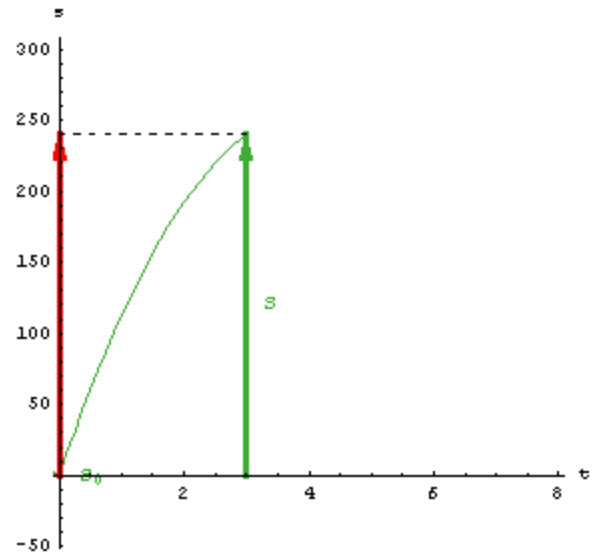
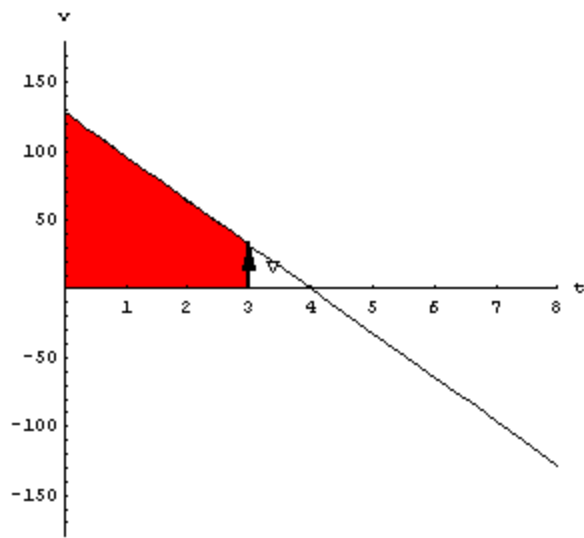


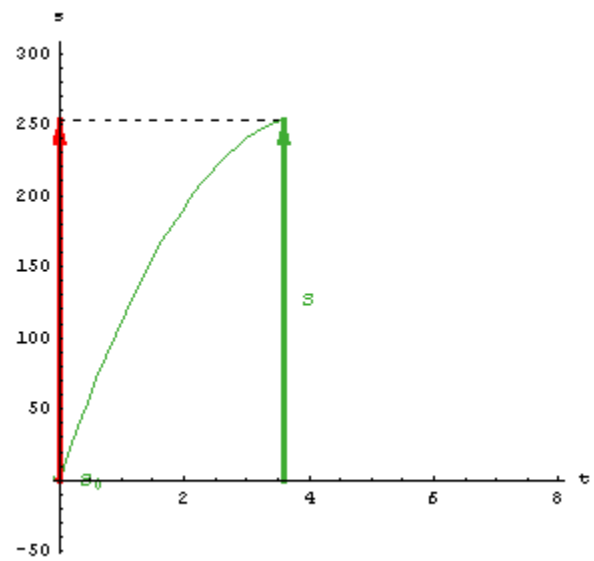
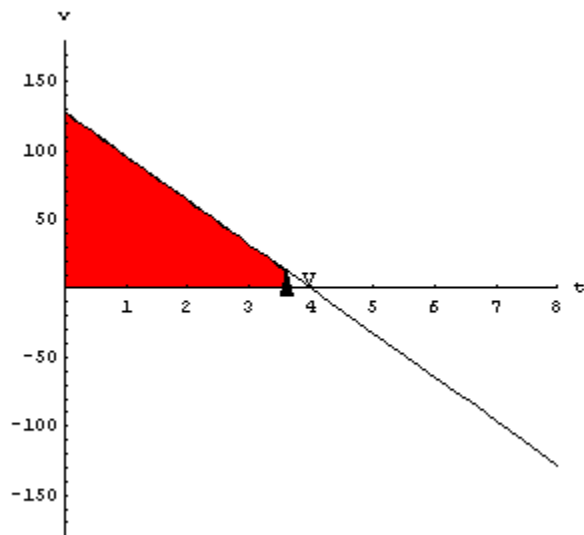
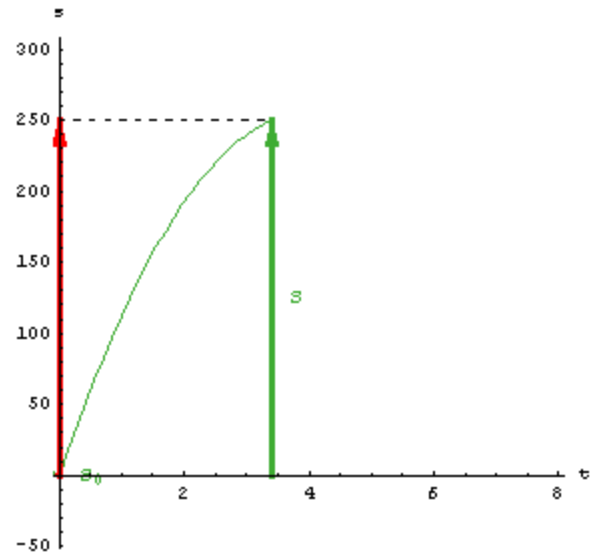
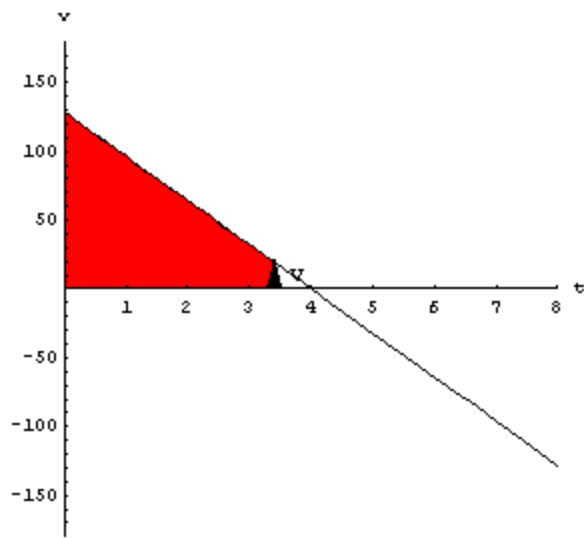


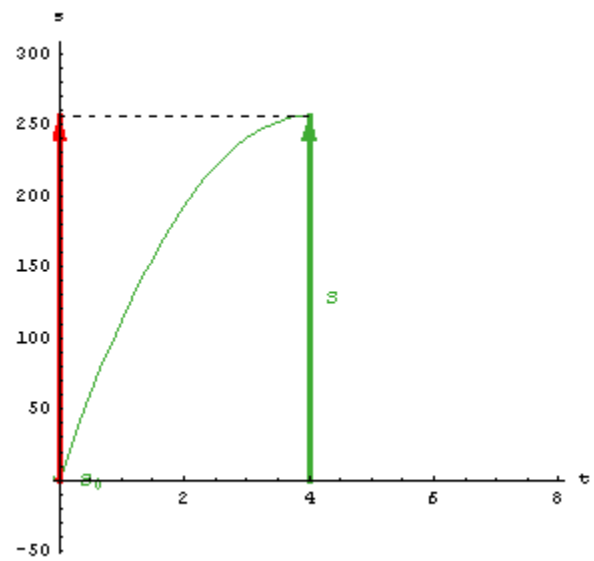
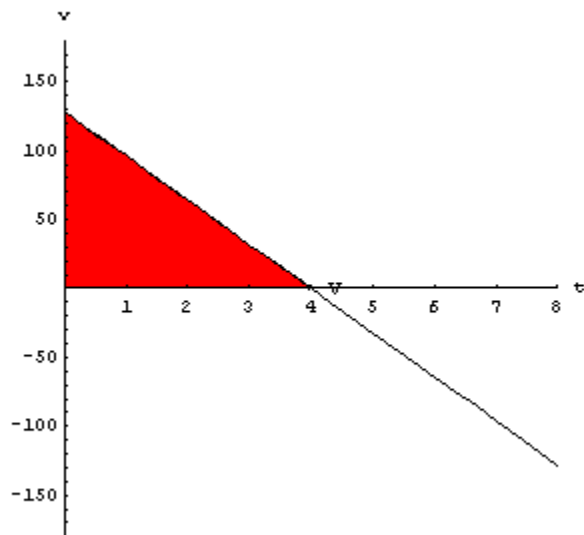
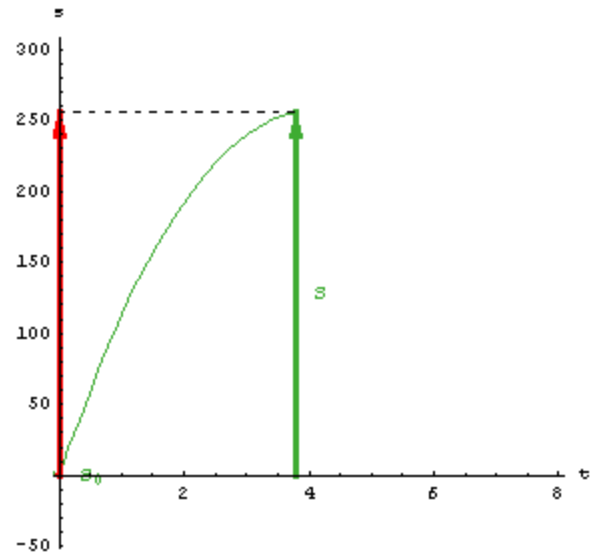
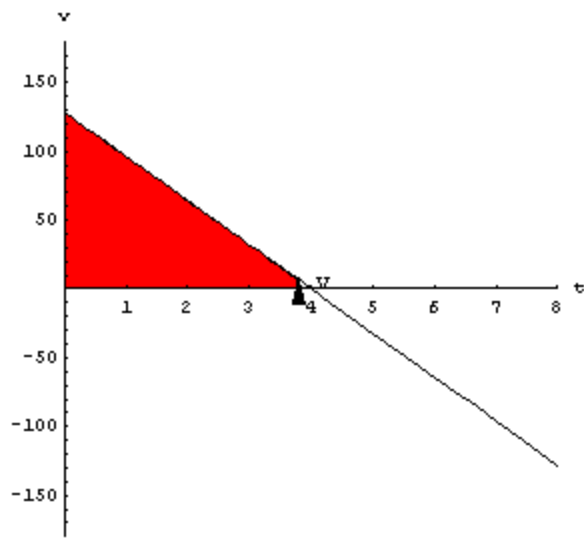


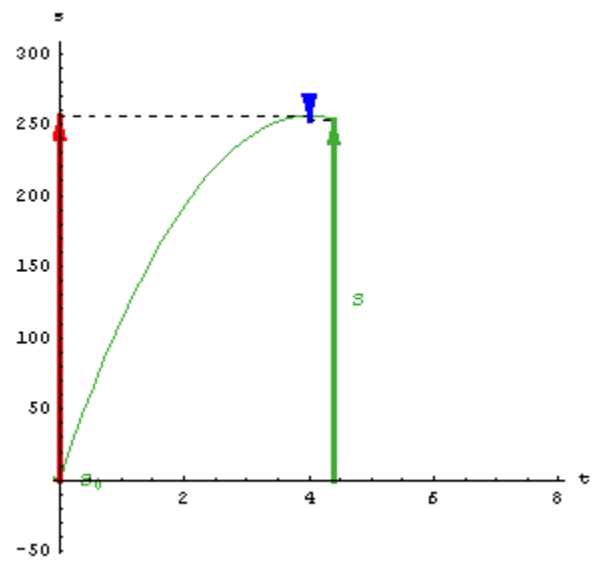
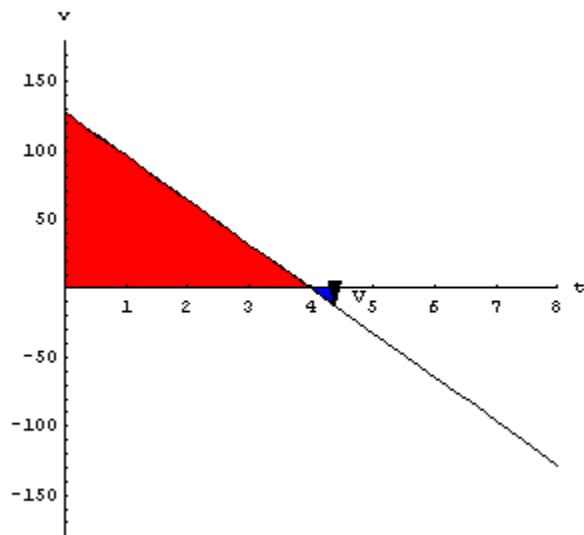
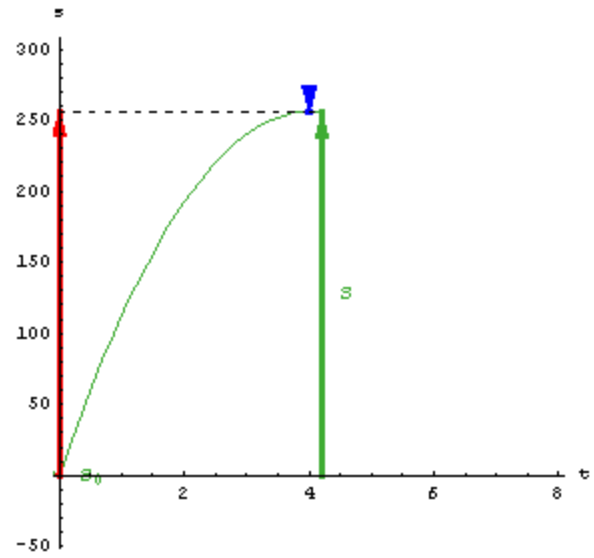
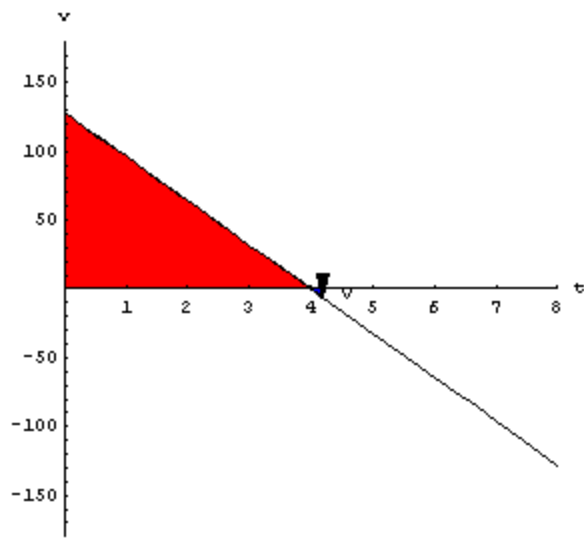


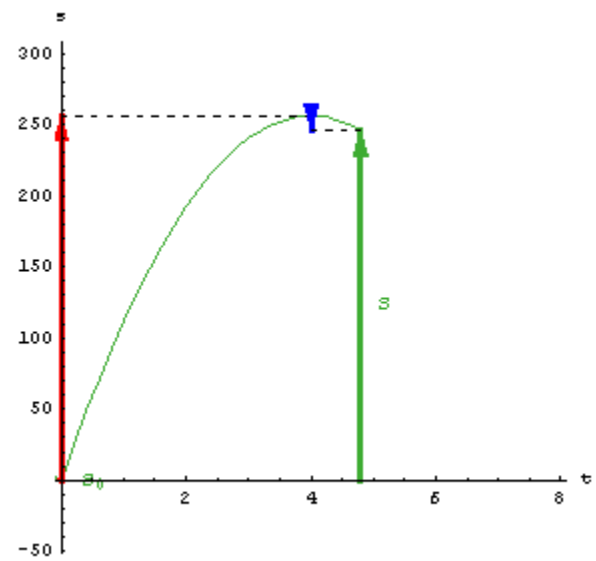
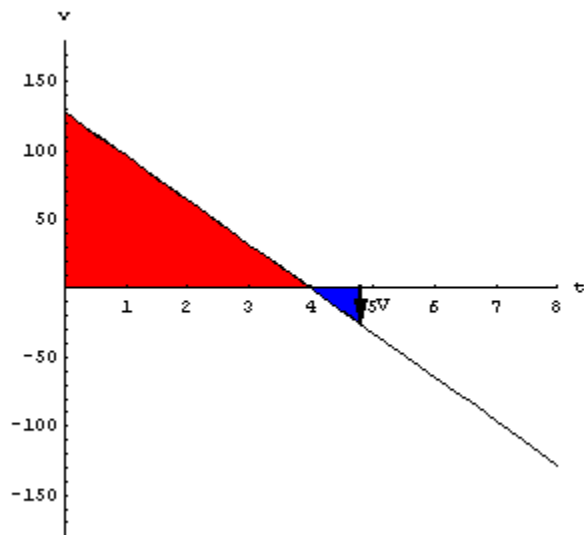
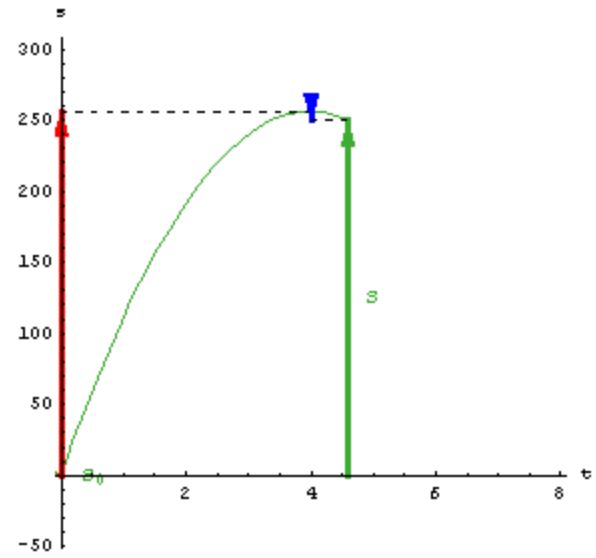
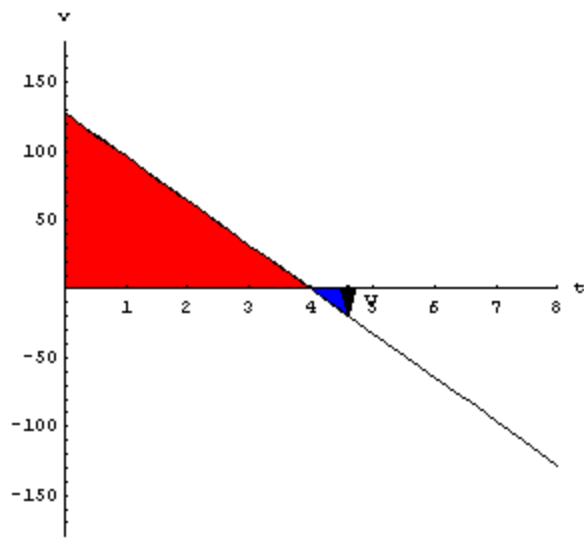


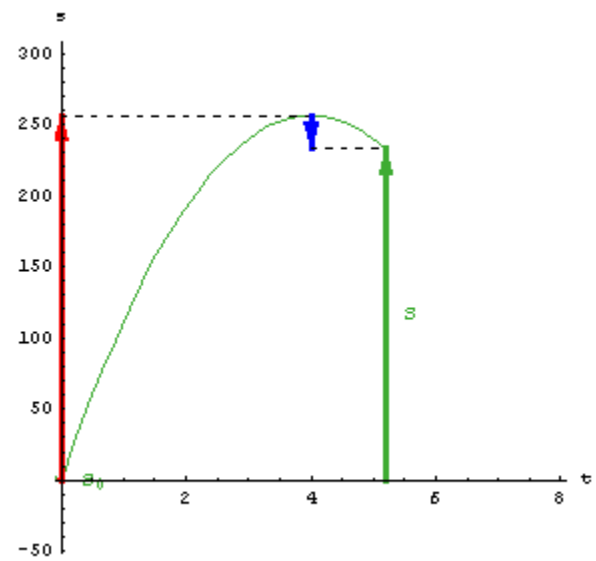
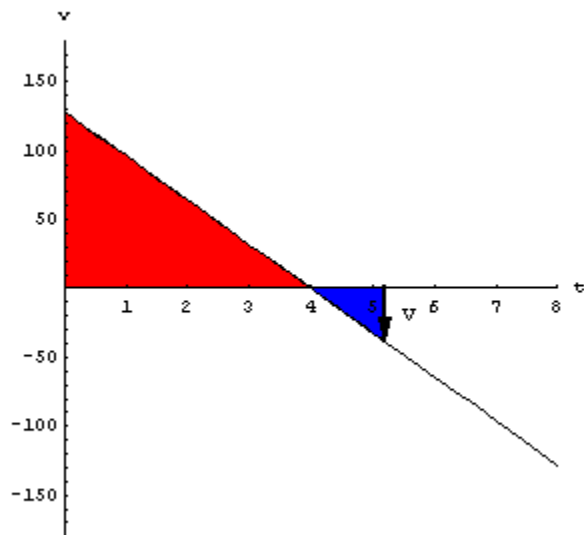
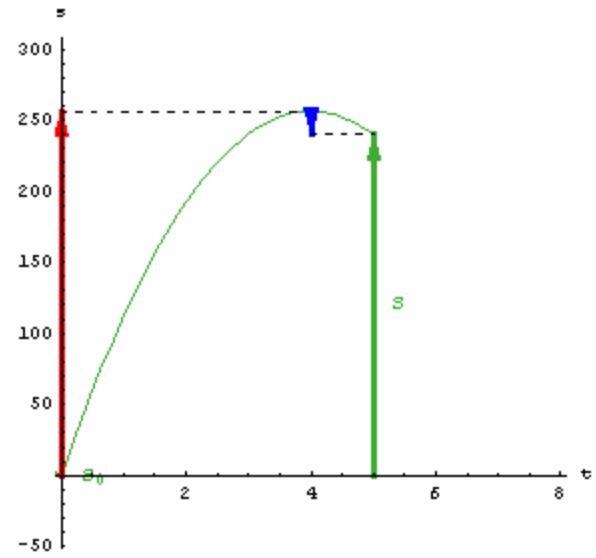
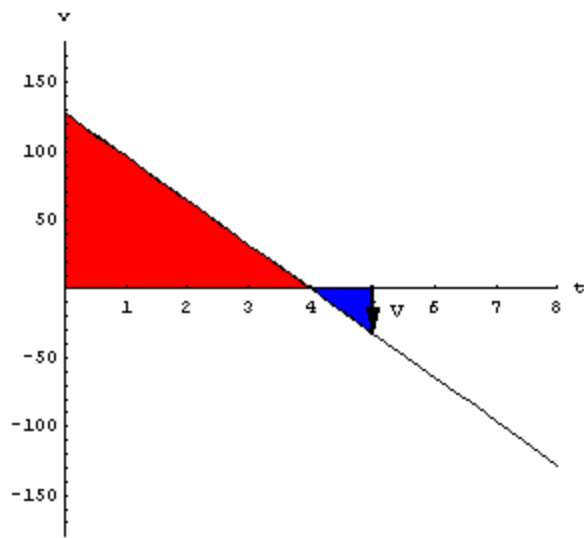


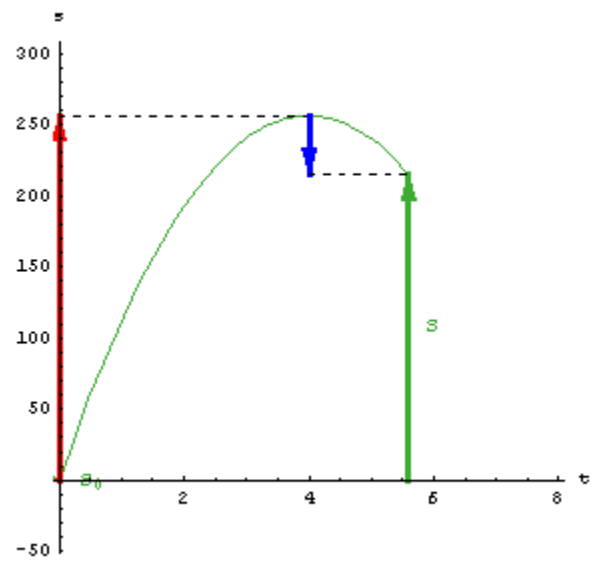
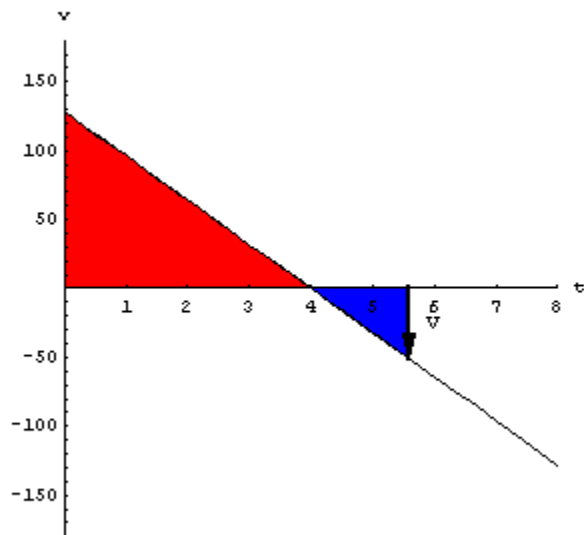
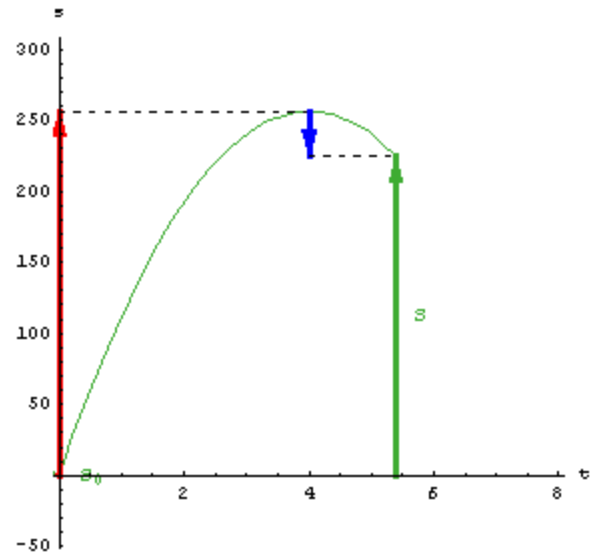
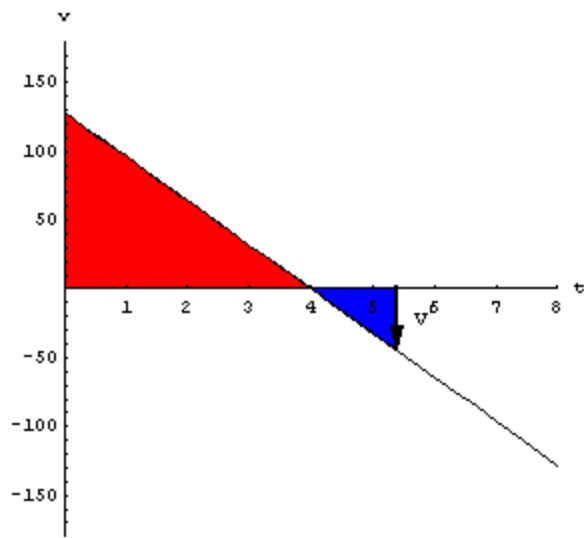


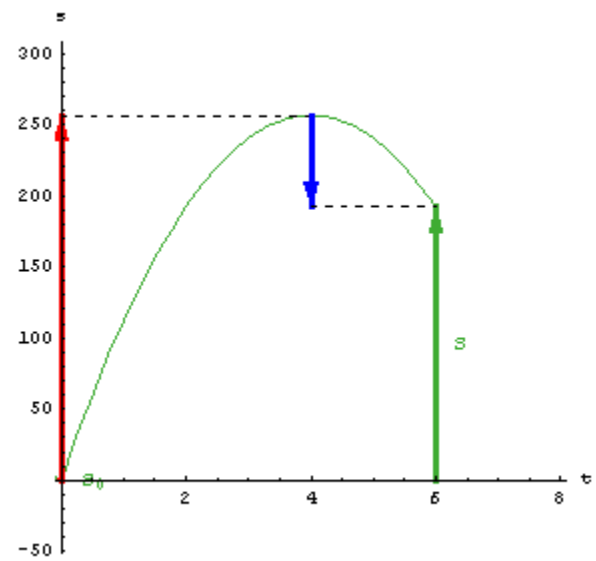
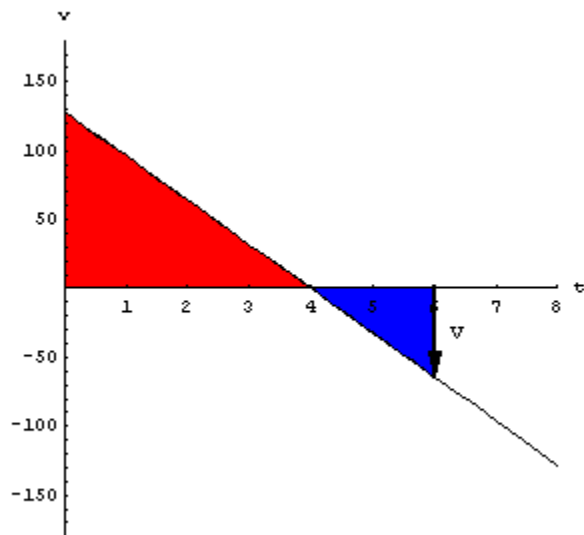
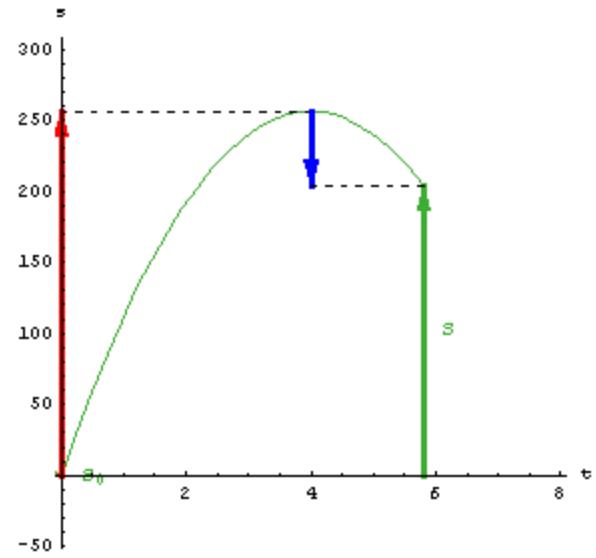
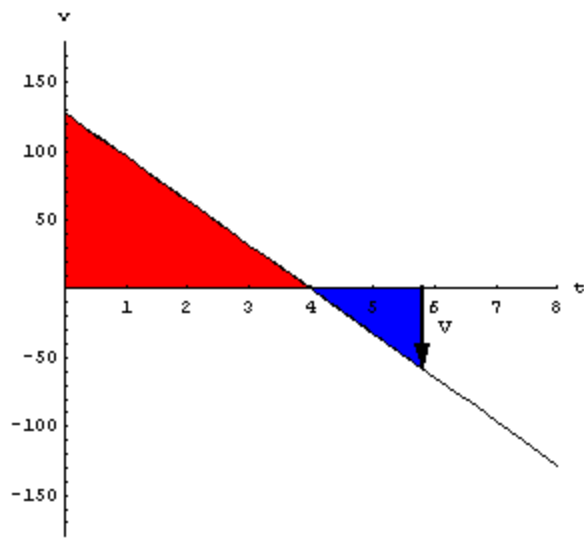


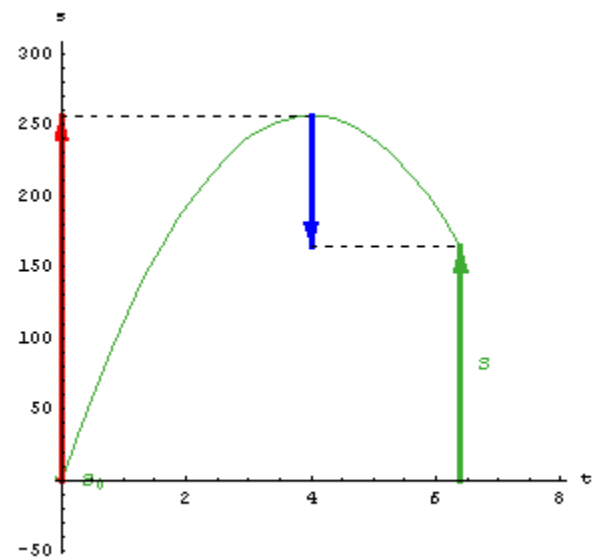
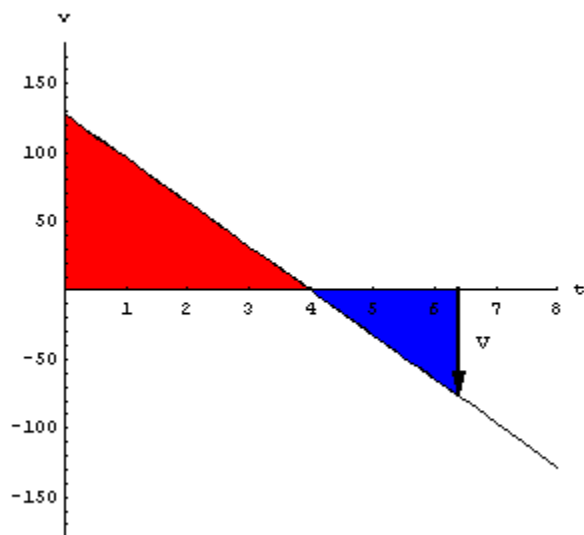
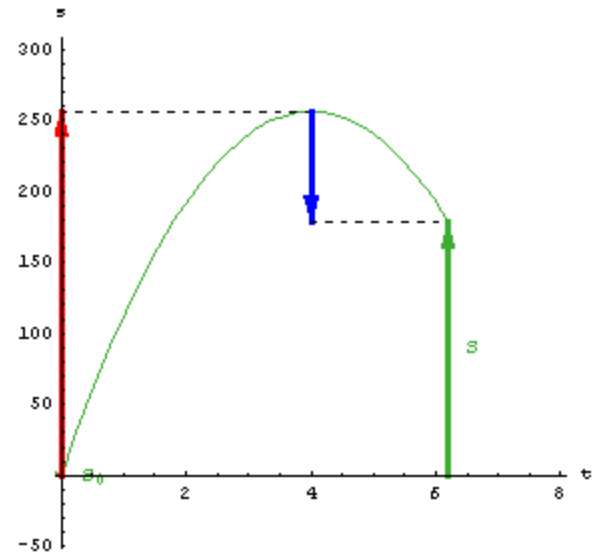
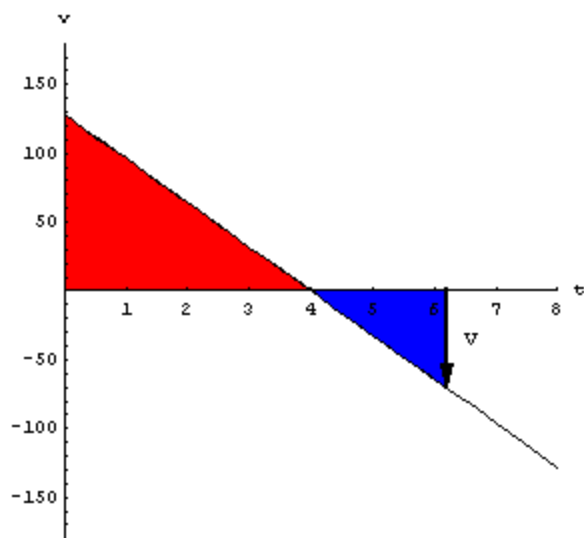


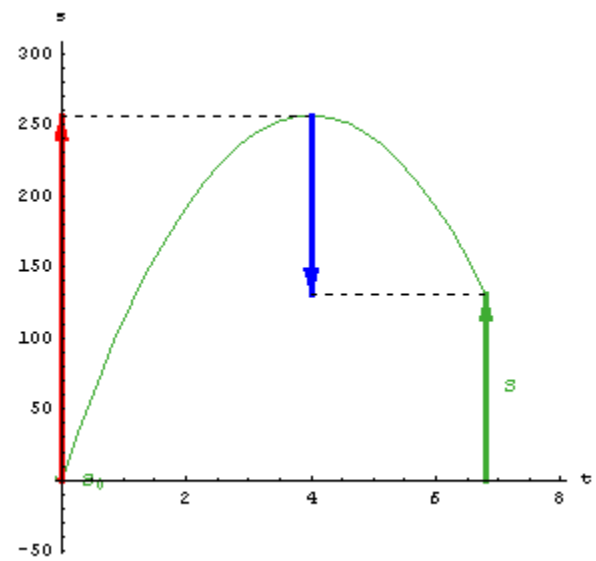
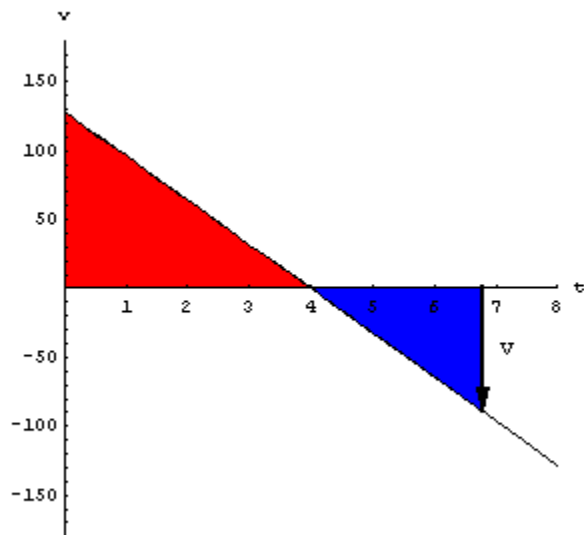
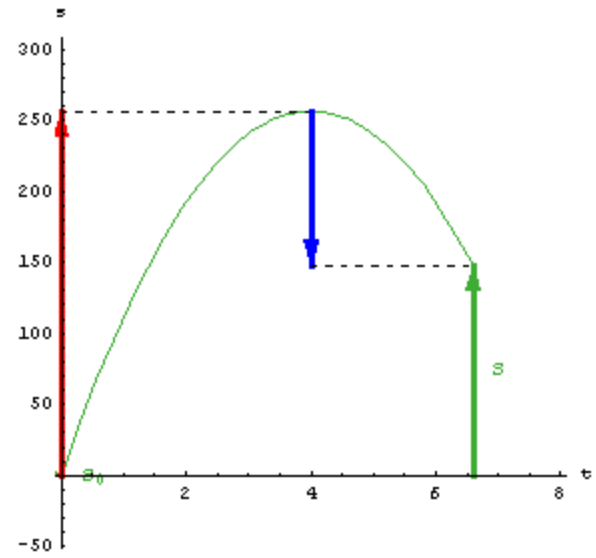
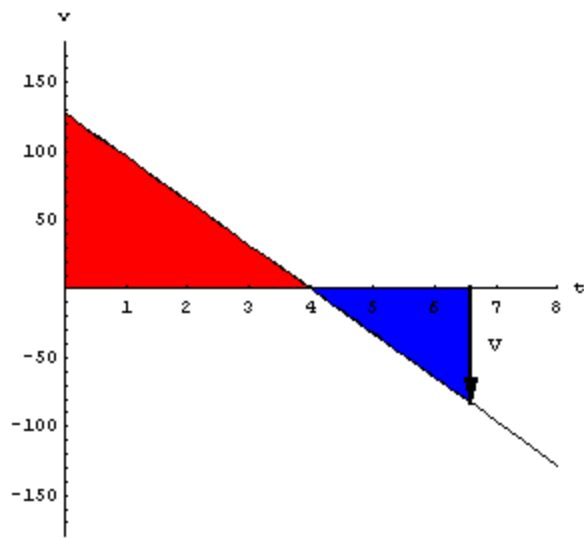


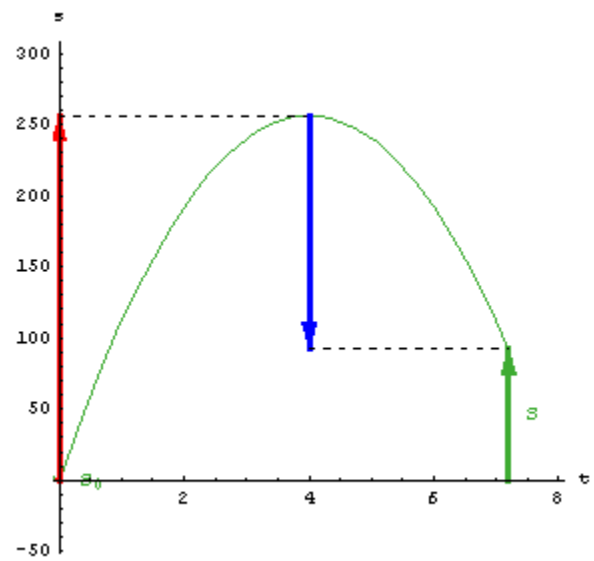
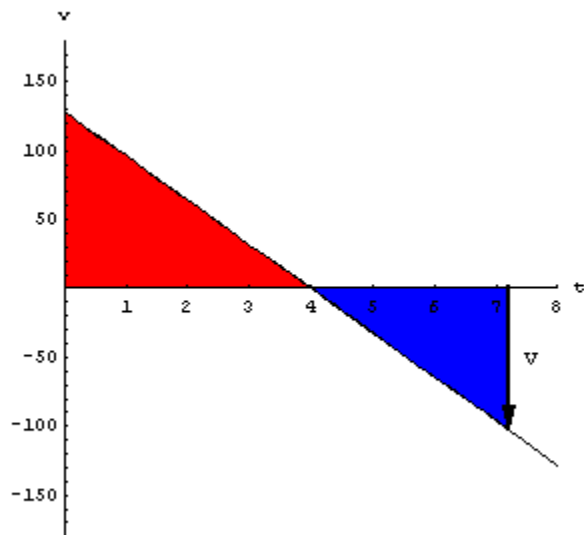
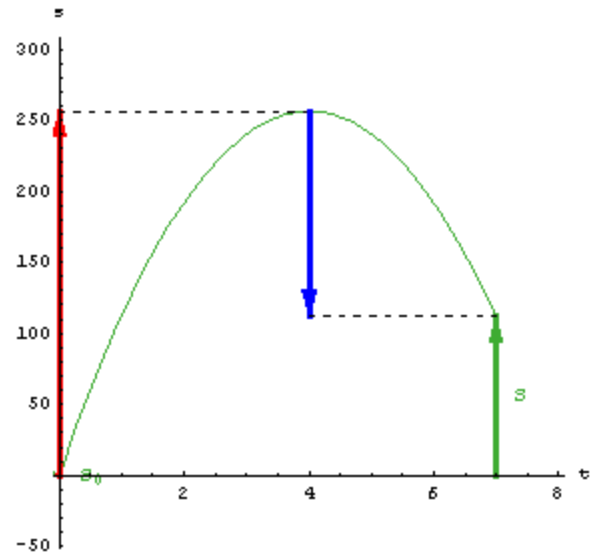
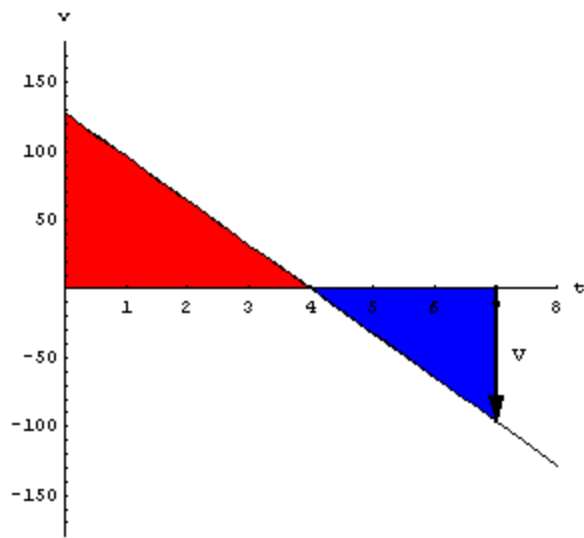


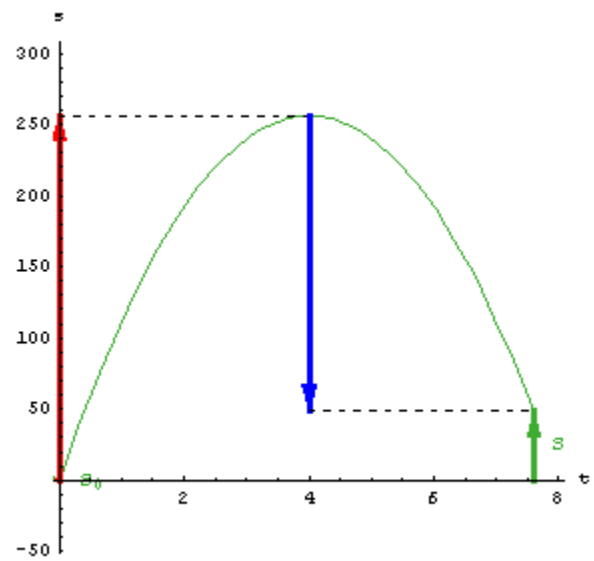
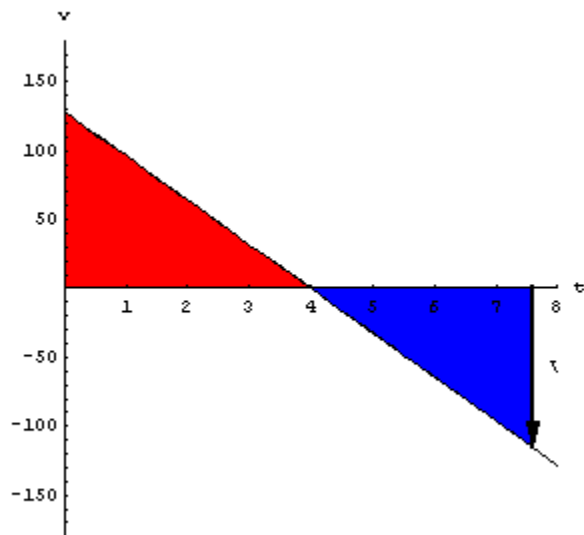
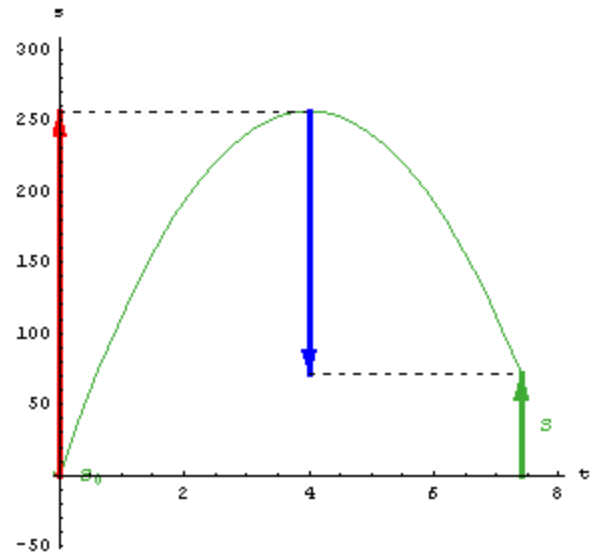
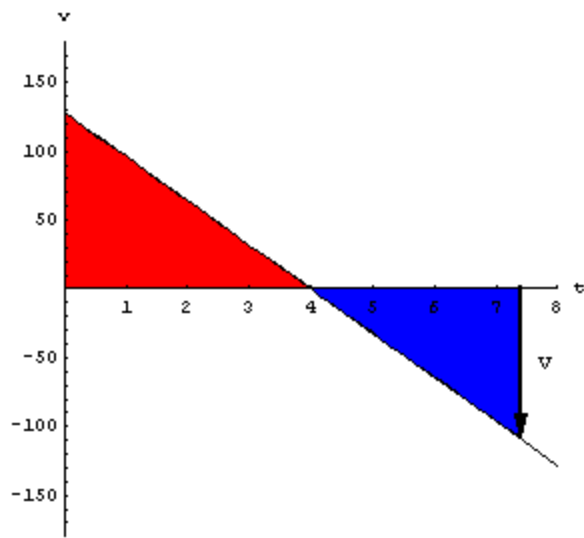


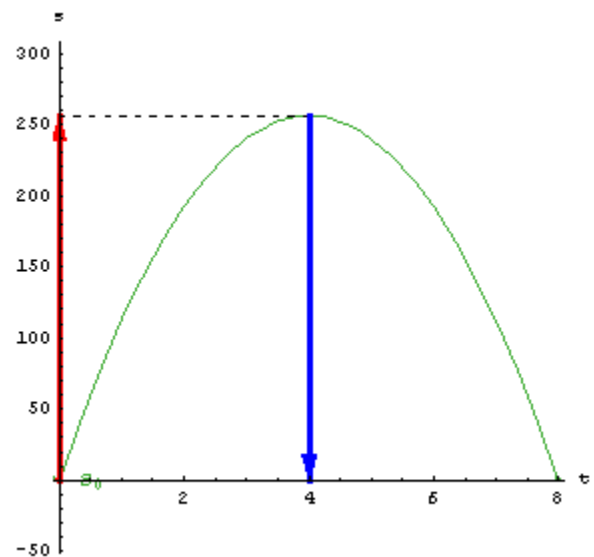
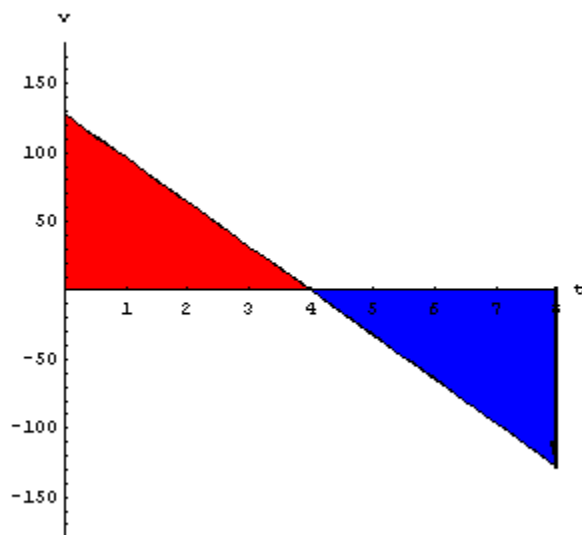
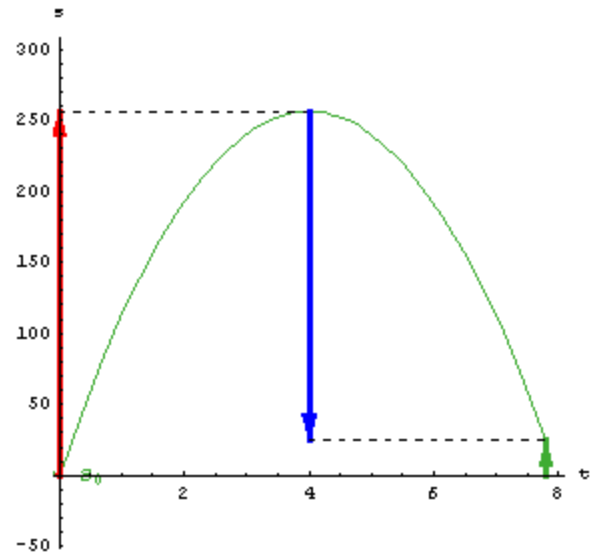
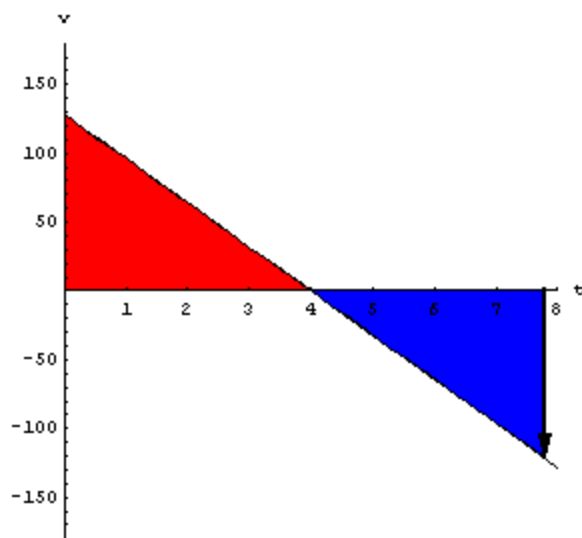












Change the commands above to try several different starting values for s_0 , highlighted in red. We suggest $s_0 = -100$ and $s_0 = 100$. What do you observe about the graphs of $s(t)$?

In the preceding example, the rate of change or the slope of the velocity function (i.e., the acceleration) is constant at $a = -g = -32 \text{ ft/sec}^2$.

Acceleration is also a vector quantity. If the acceleration is in the negative direction, the velocity (which can be positive or negative) is decreasing with time. And, if the acceleration is

in the positive direction, the velocity (which can be positive or negative) is increasing with time. If the velocity and the acceleration have the same sign (positive or negative), then the object is speeding up, and if they have opposite signs, then the object is slowing down.

The specially designed **velocity[]** command illustrates this idea by producing a sequence of graphs that show the accumulation of velocity in two different formats. The first format shows the changes in velocity as shaded areas between the acceleration function and the t -axis, and the second format shows the velocity changes as vertical increments of the function on the velocity graph.

Red areas on the acceleration graph are numerically equal to increases in the velocity due to positive acceleration, whereas blue areas are equal to decreases in velocity due to negative acceleration. Similarly, the lengths of the red arrows on the velocity graph represent increases in the velocity function, and the lengths of the blue arrows represent decreases in the velocity. The lengths of the red arrows on the velocity function are numerically equal to the corresponding red areas on the acceleration graph, and the lengths of the blue arrows on the velocity function are numerically equal to the corresponding blue areas on the acceleration graph.

In this case, the initial value of the velocity is +128 ft/sec.

In[39]:=

```
Clear[a];
```

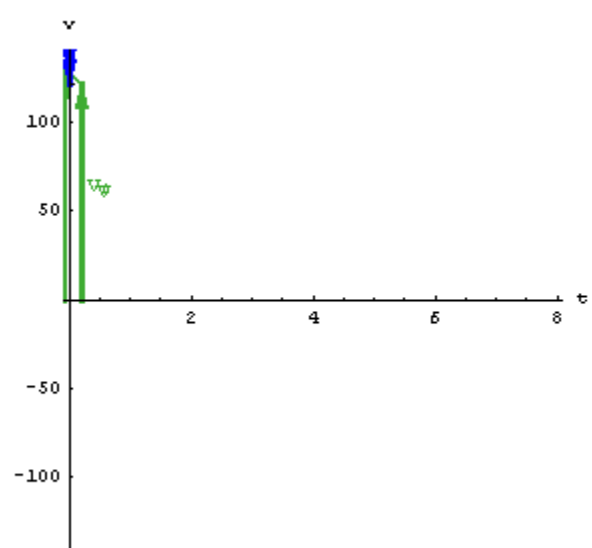
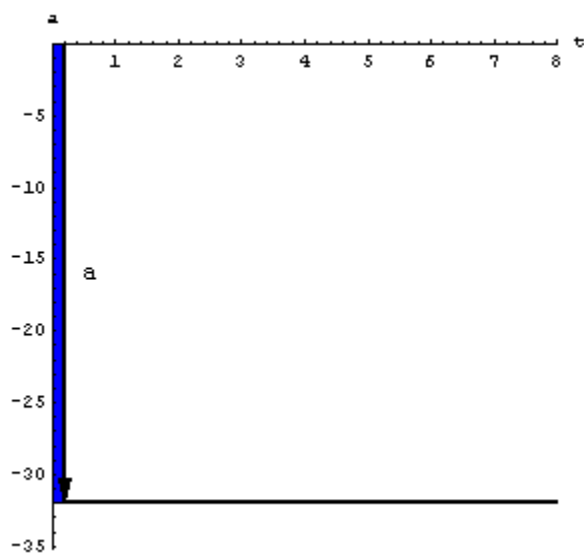
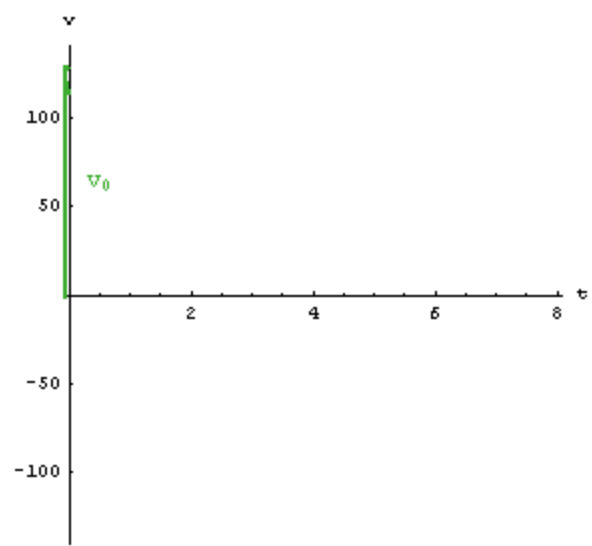
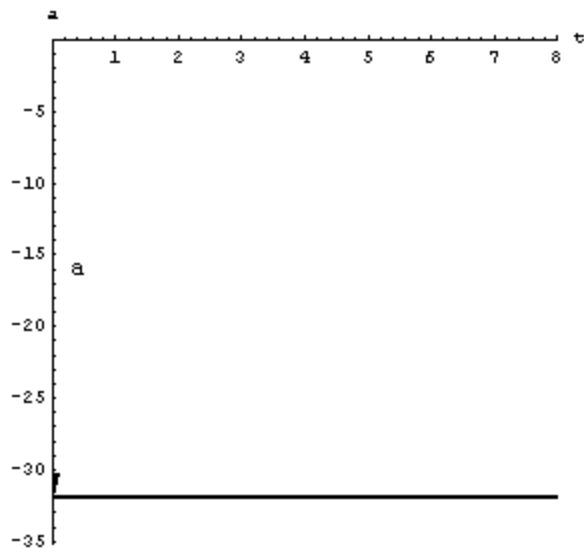
```
a[t_] = -32;
```

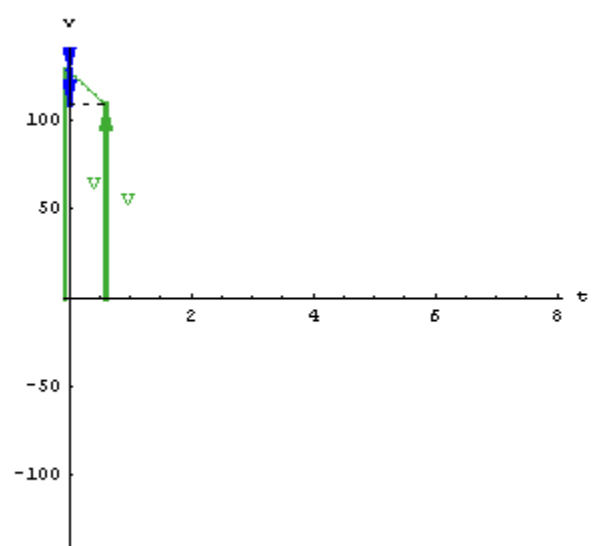
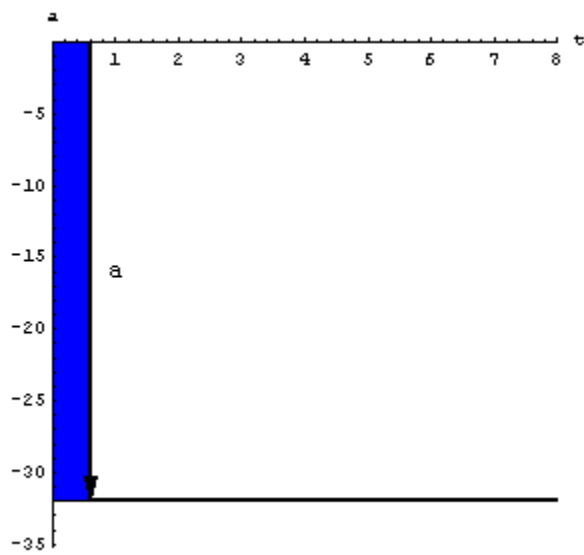
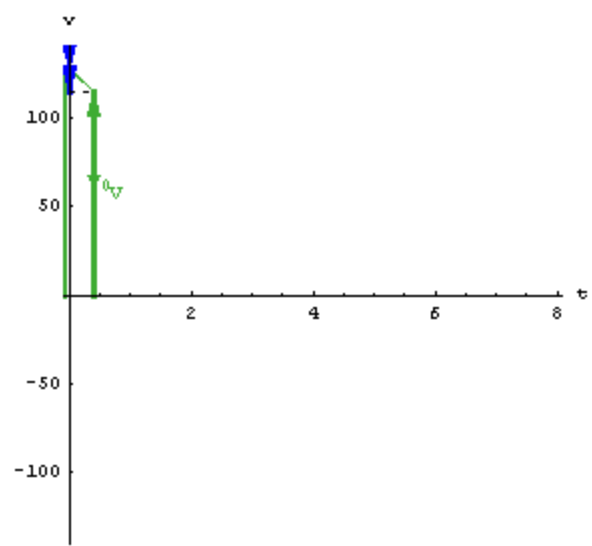
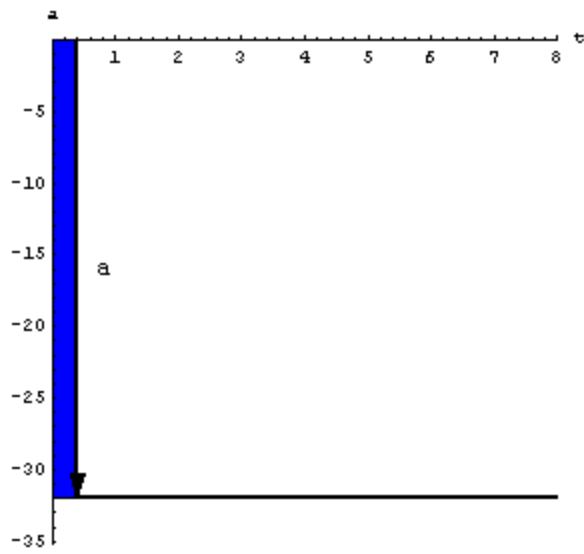
```
v0 = 128;
```

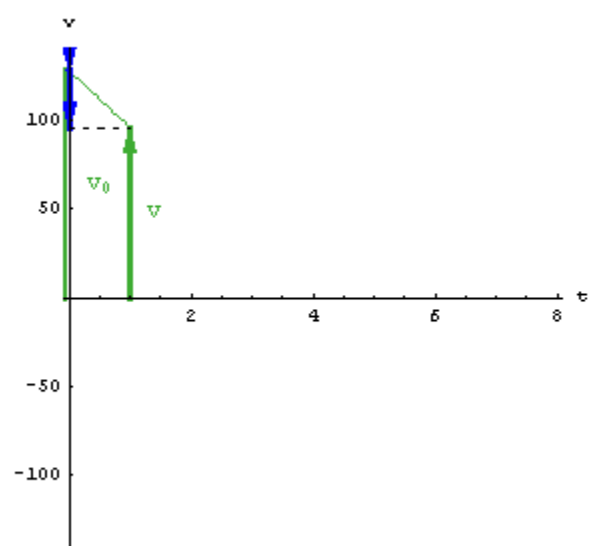
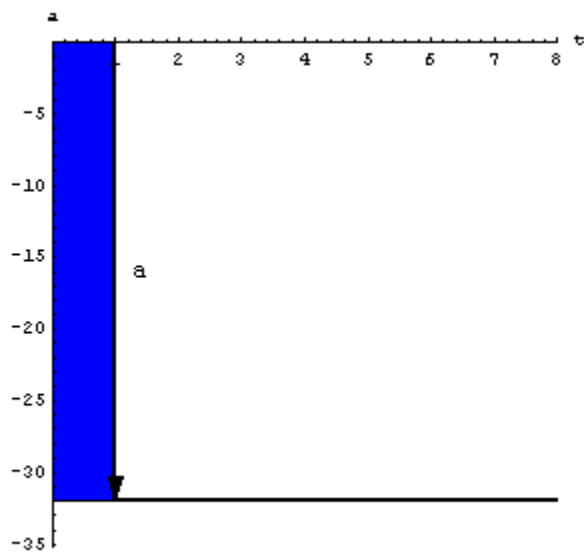
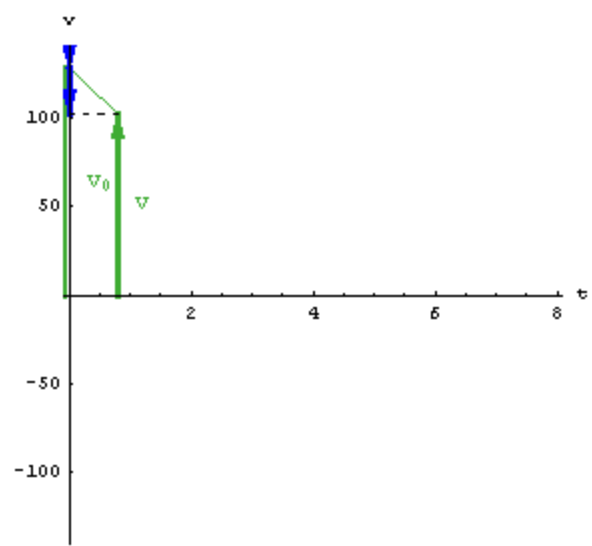
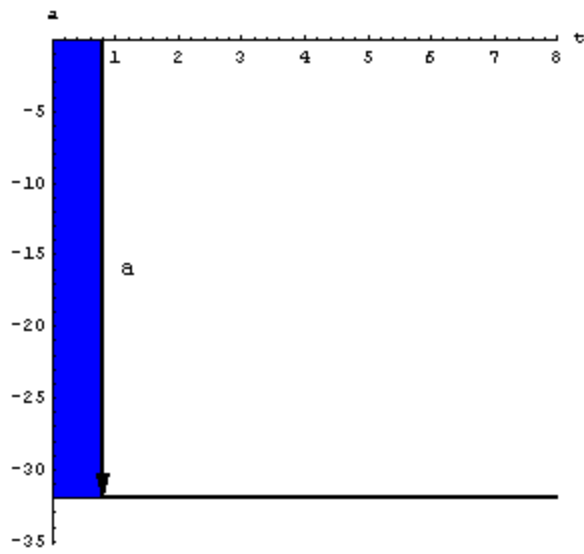
Note: The next command generates a lot of graphics, which may fill up your computer's memory. Before executing the next command, pull down the Kernel menu and select Delete All Output. All computed values are saved when you do this.

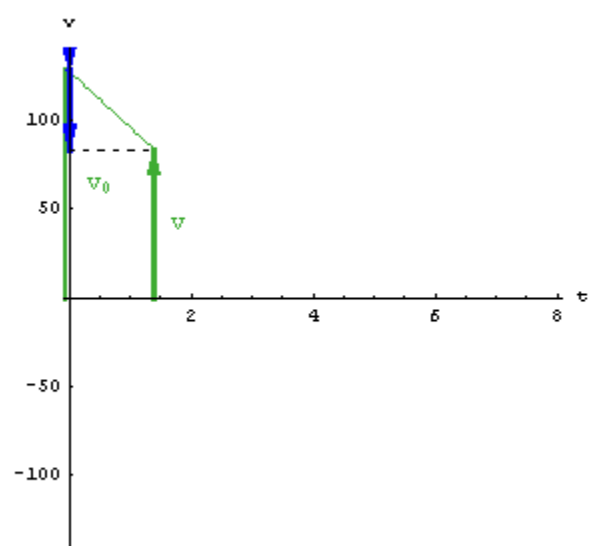
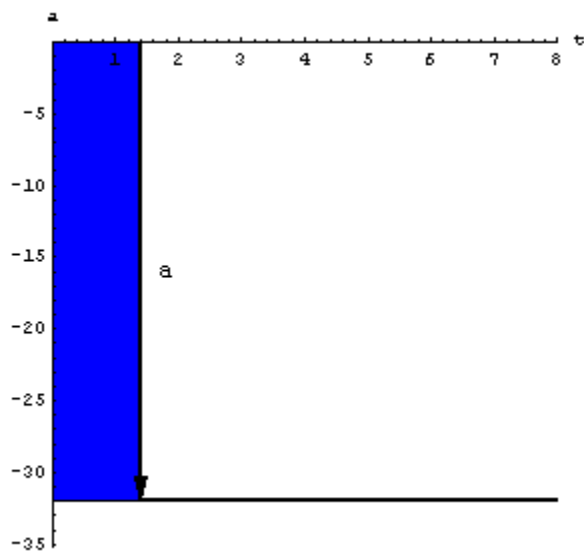
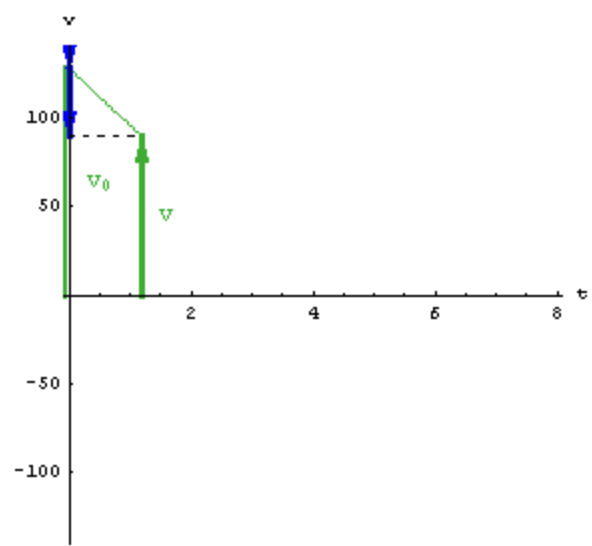
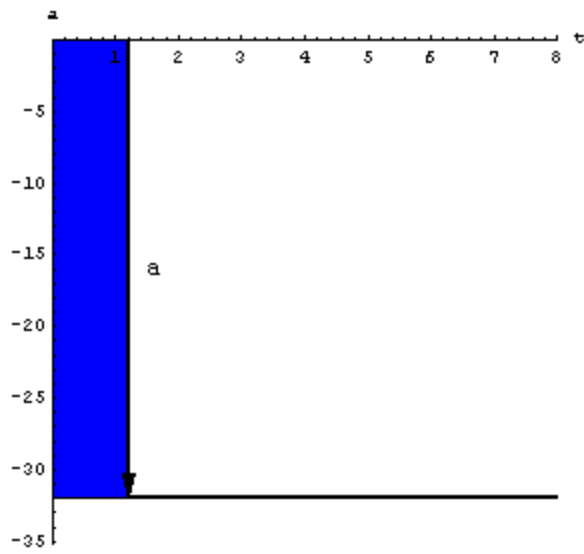
In[42]:=

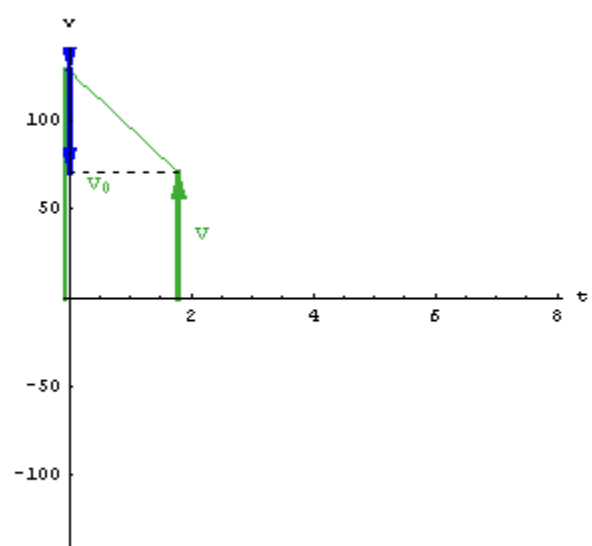
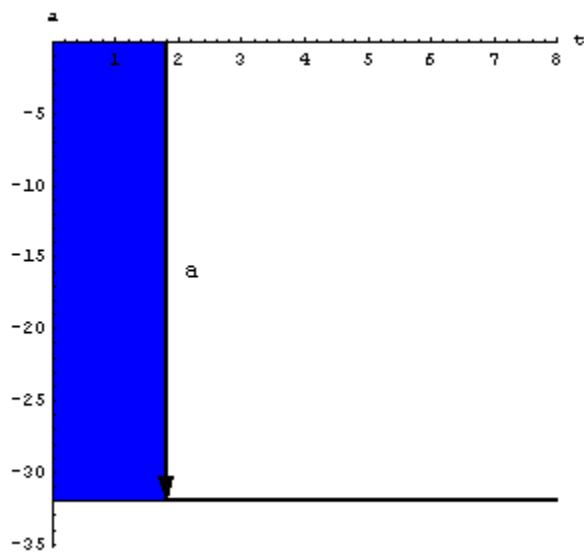
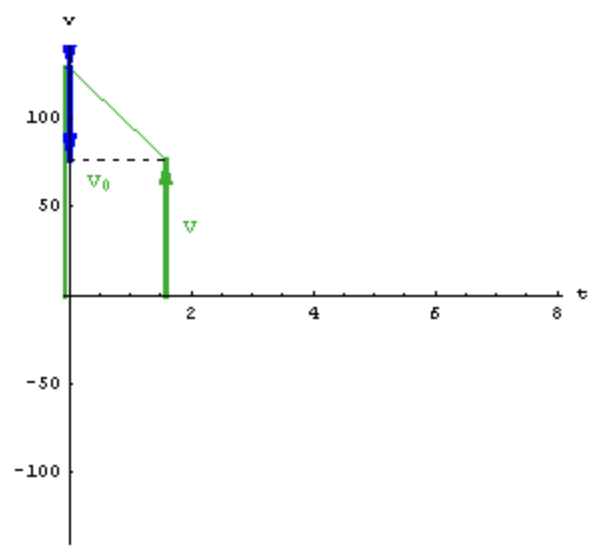
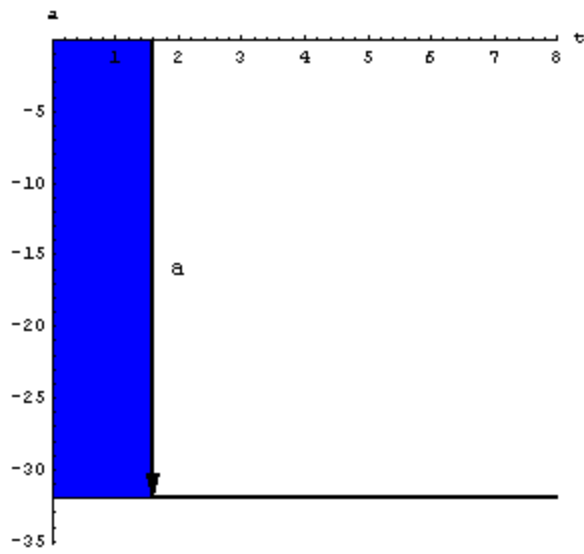
```
velocity[a[t], {t, 0, 8}, v0];
```

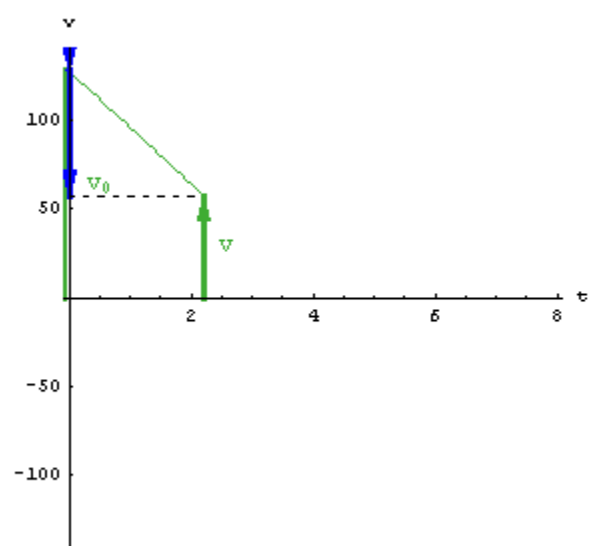
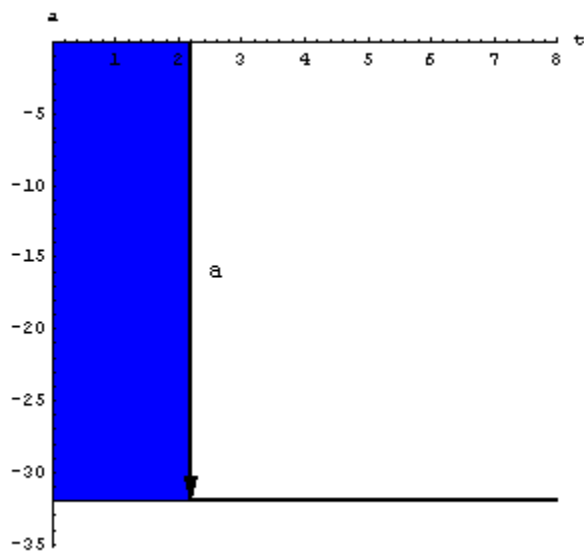
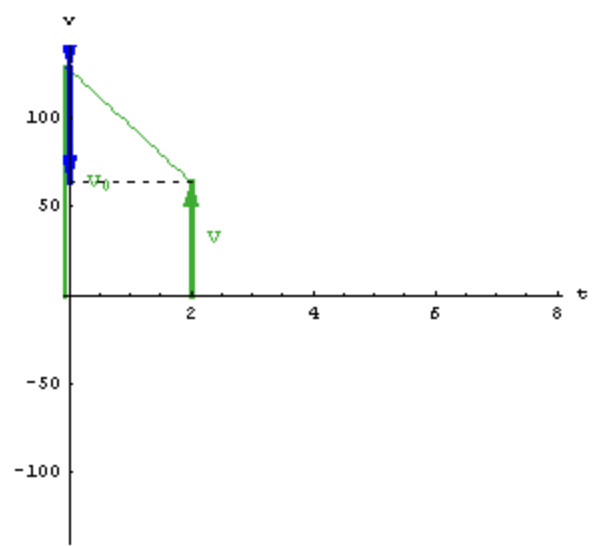
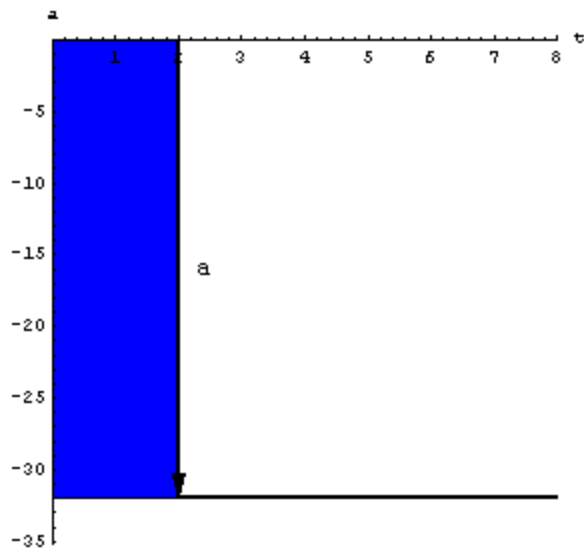


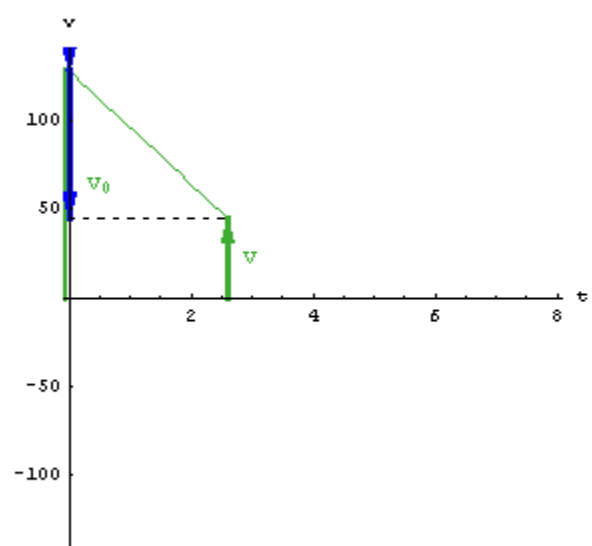
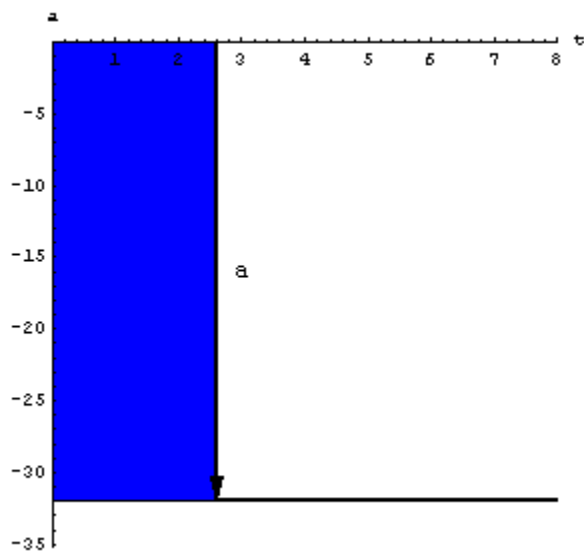
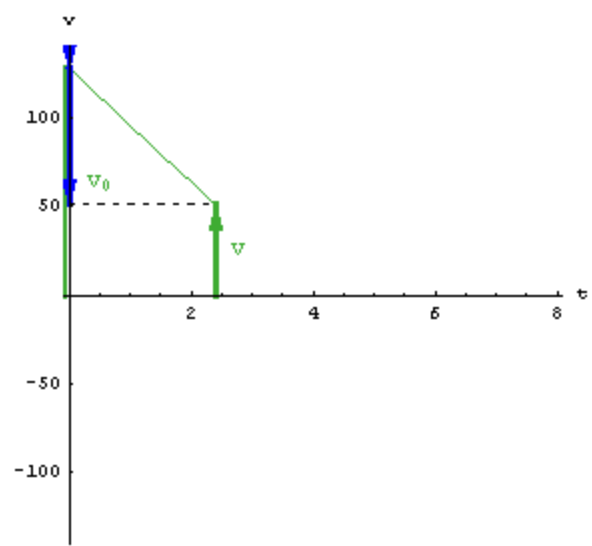
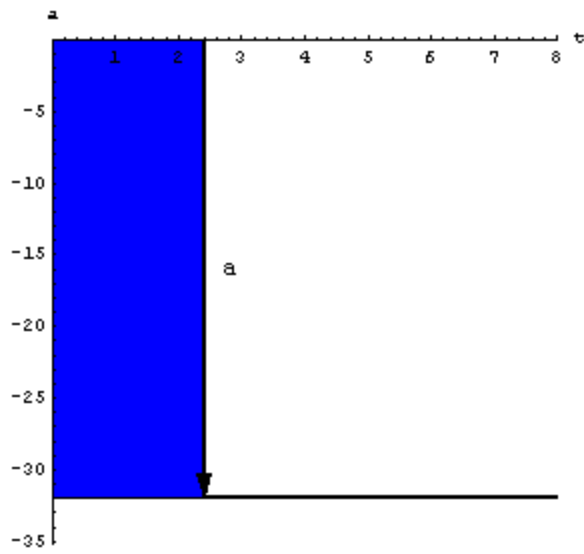


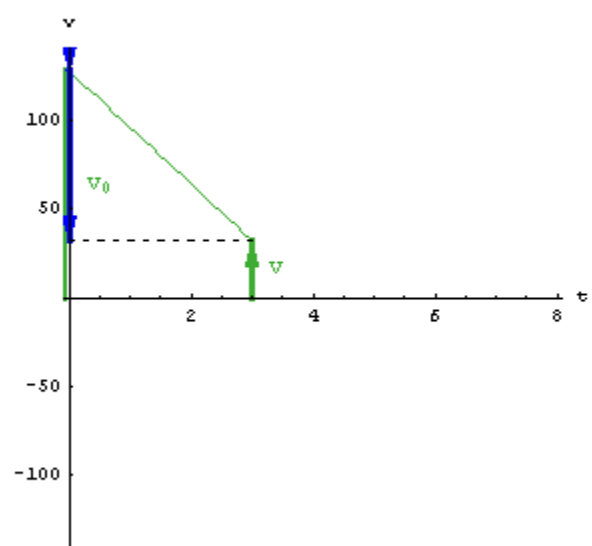
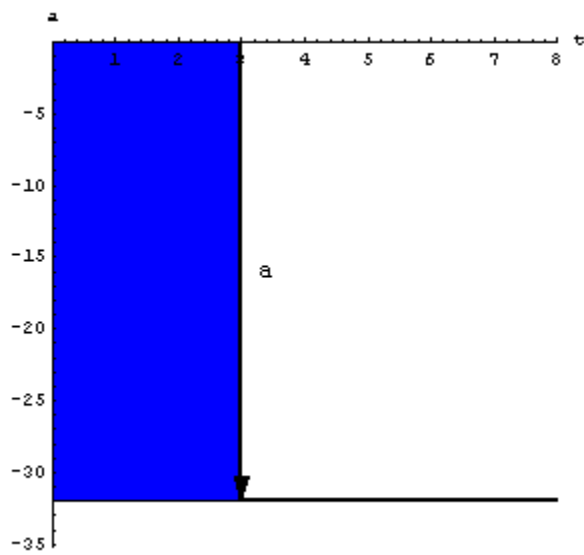
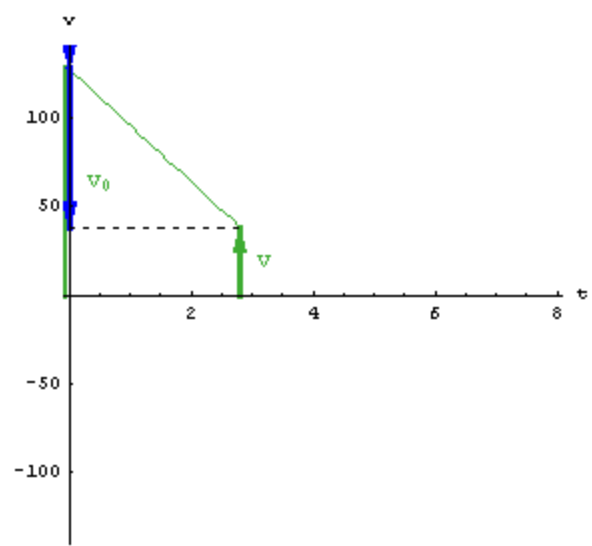
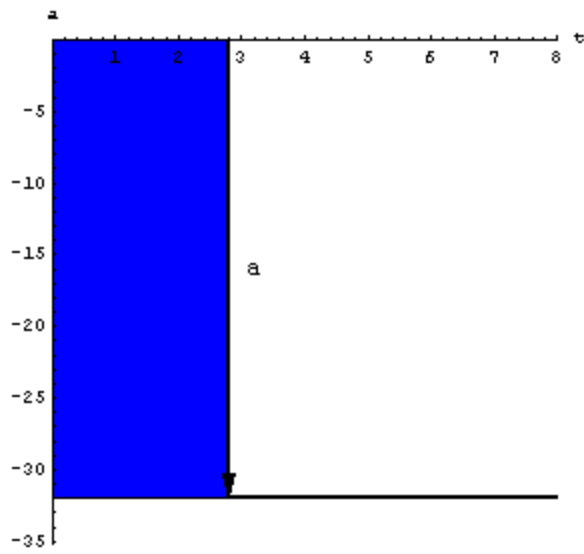


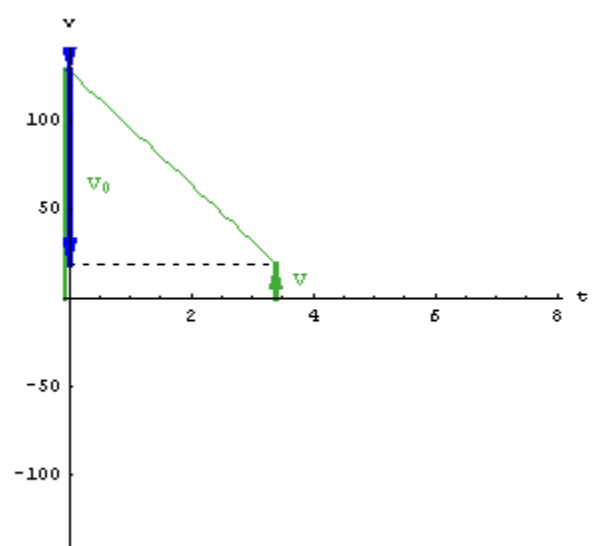
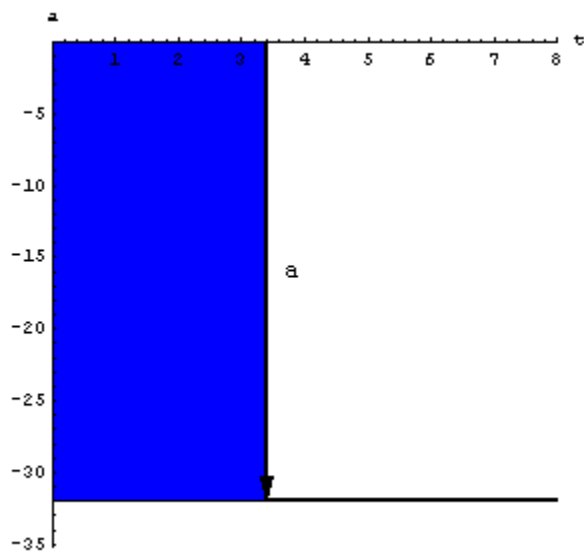
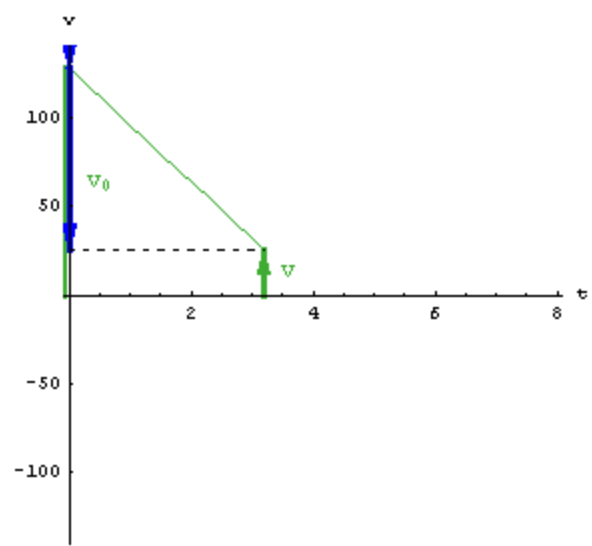
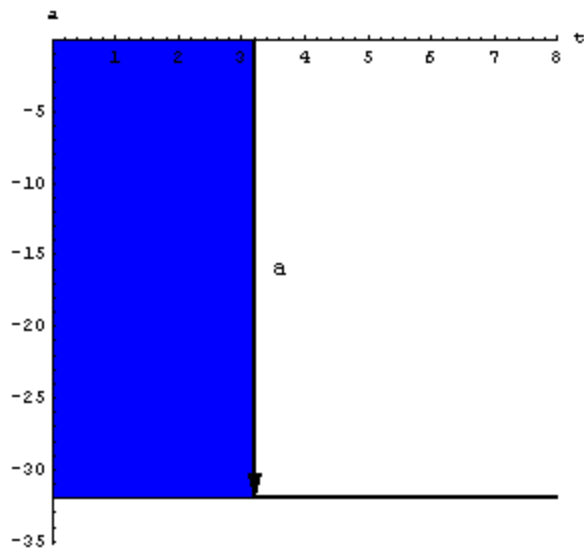


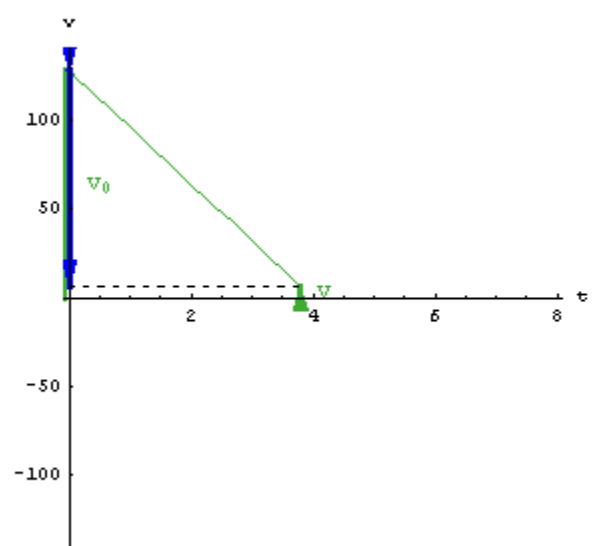
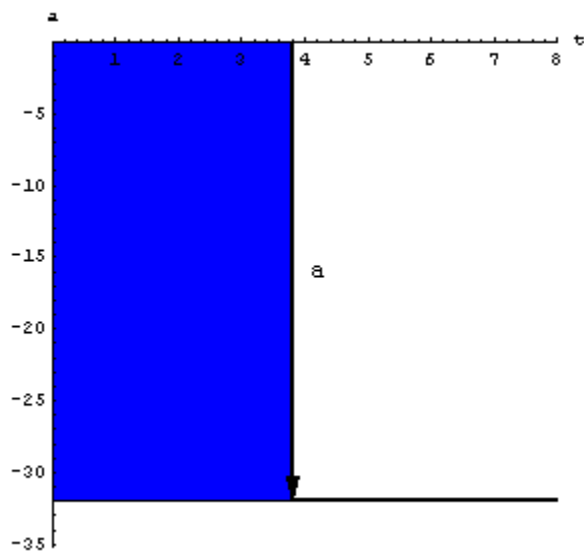
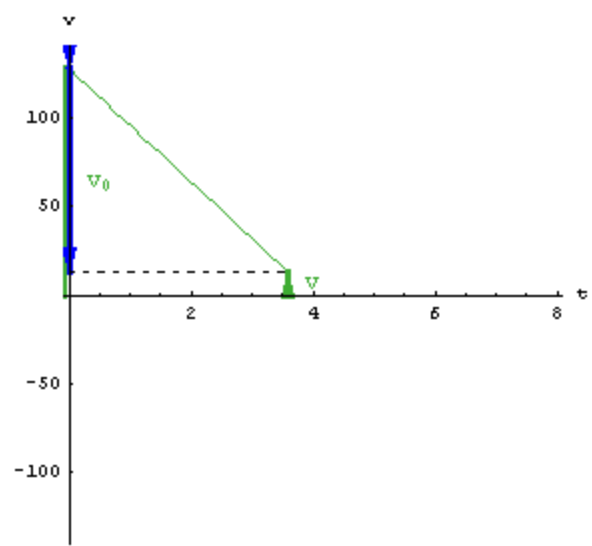
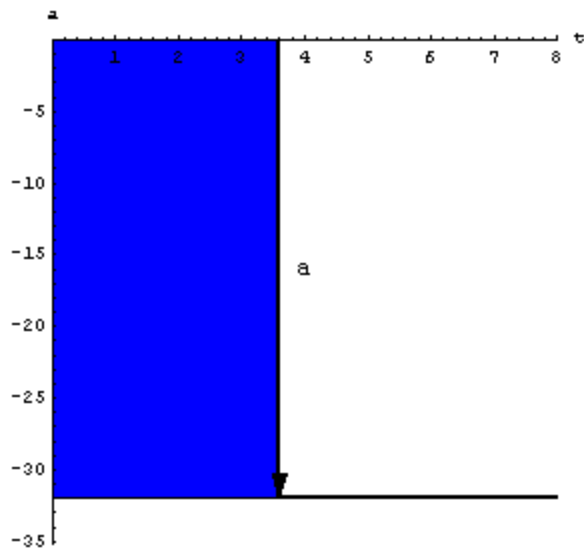


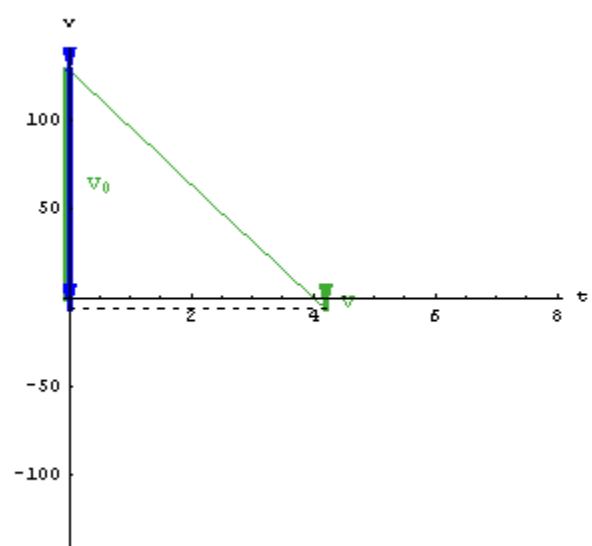
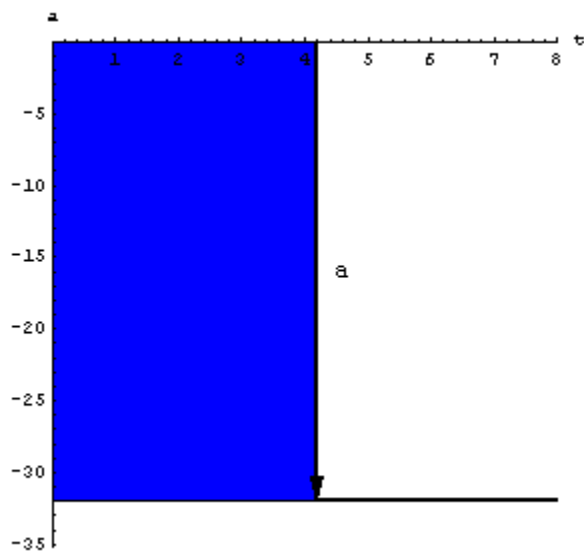
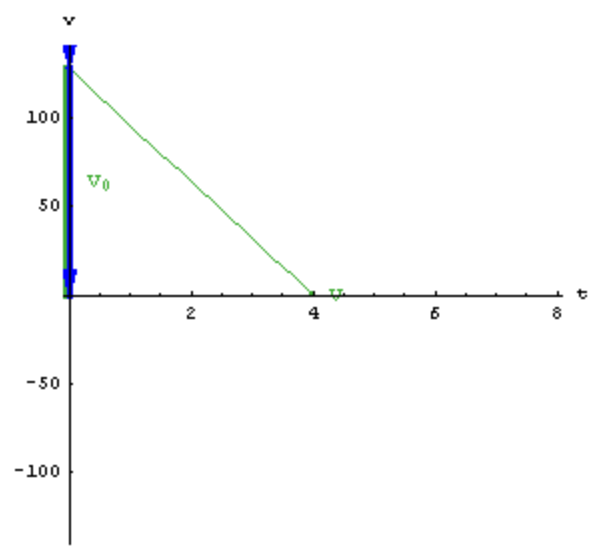
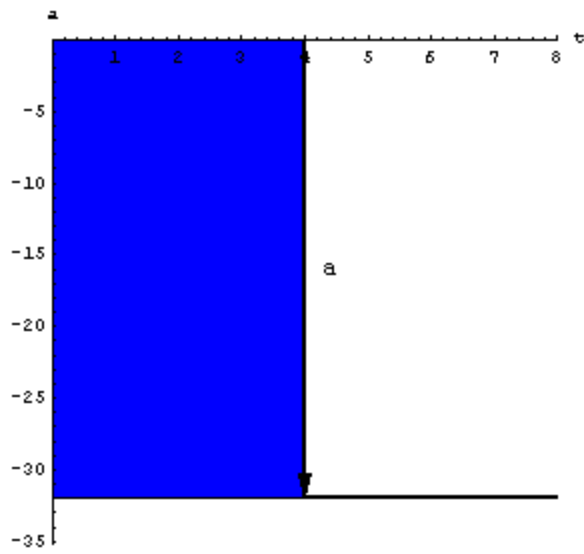


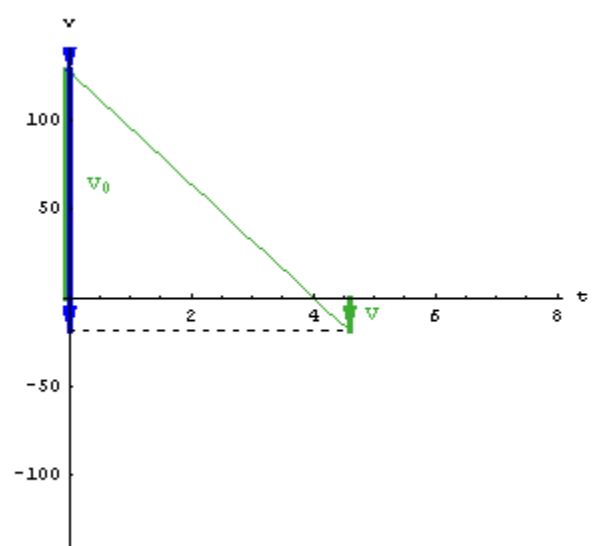
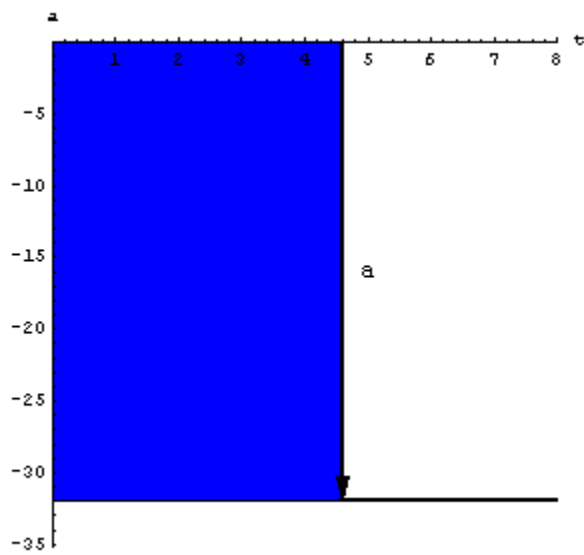
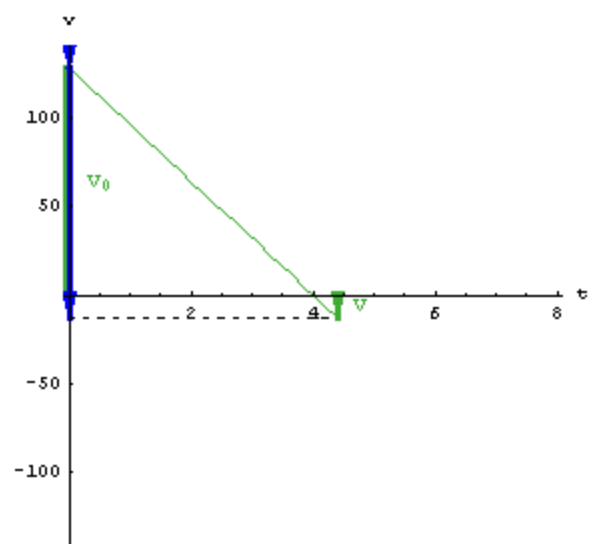
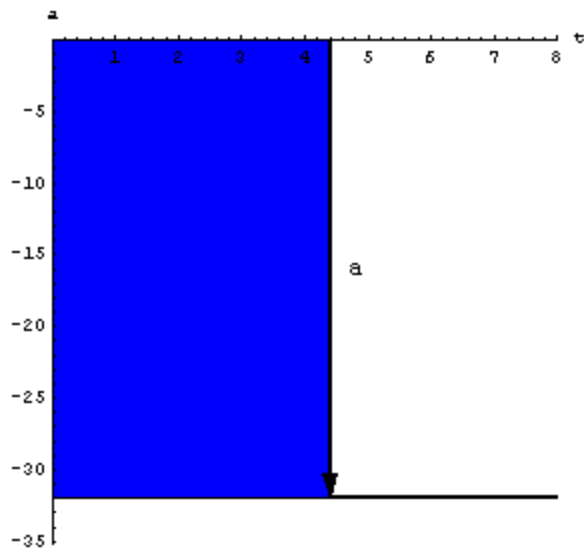


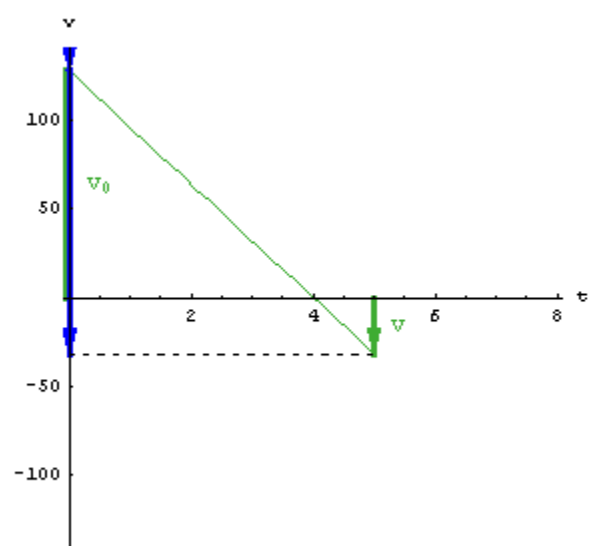
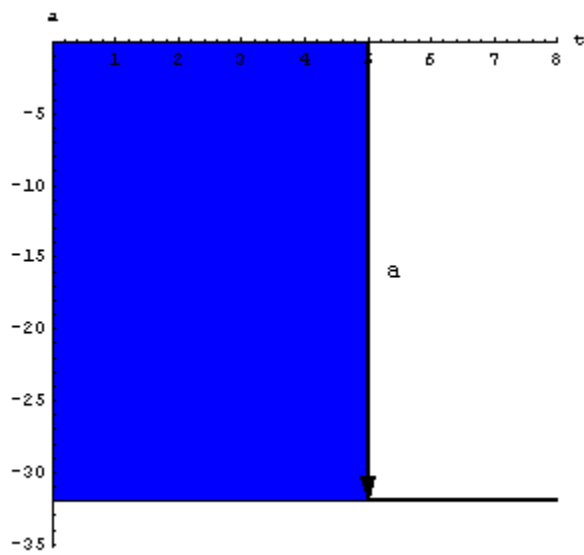
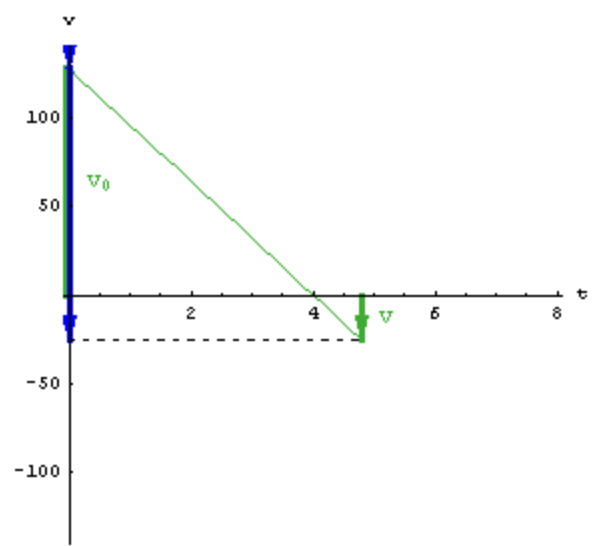
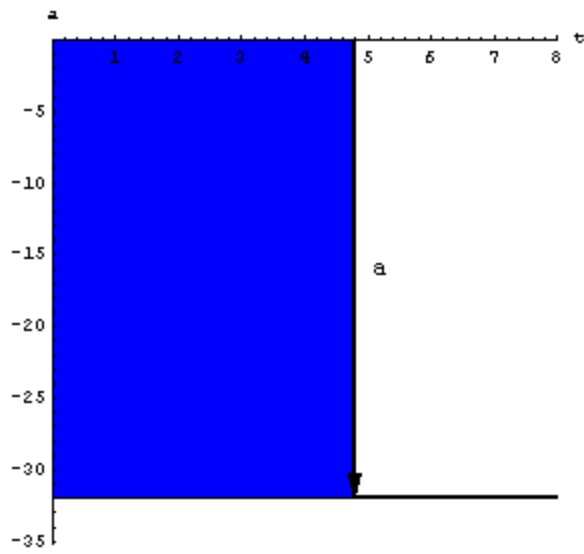


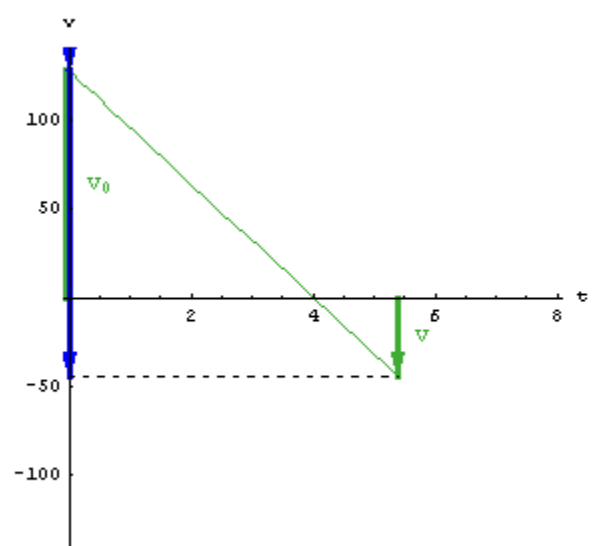
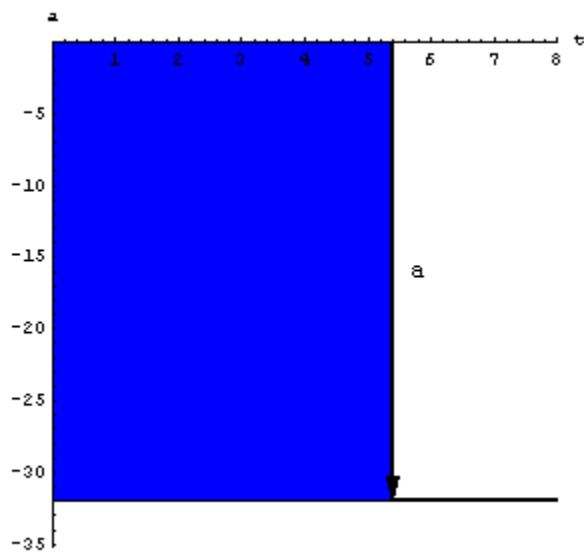
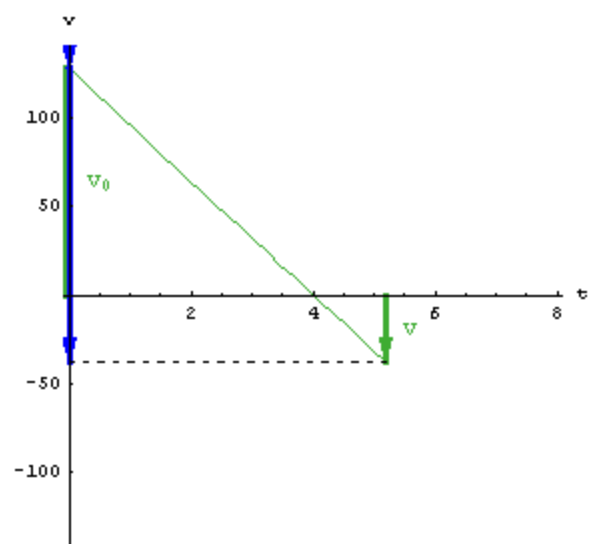
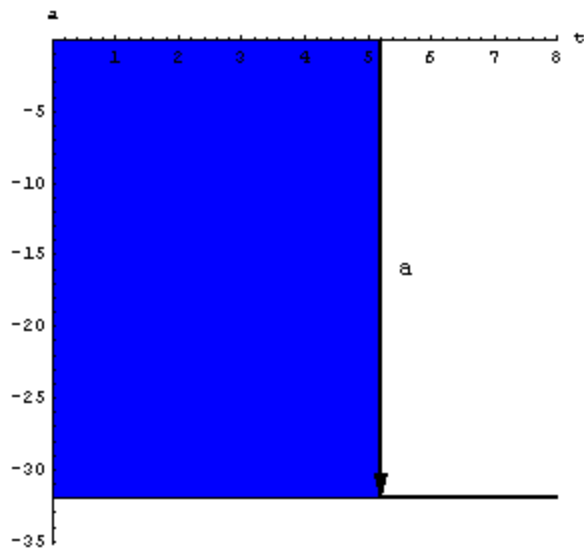


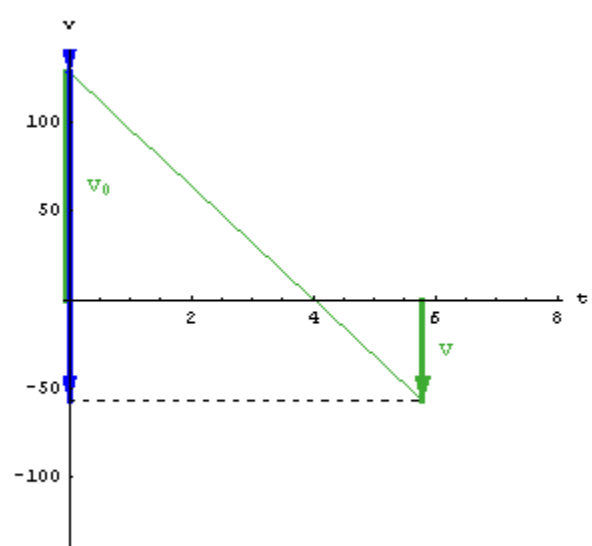
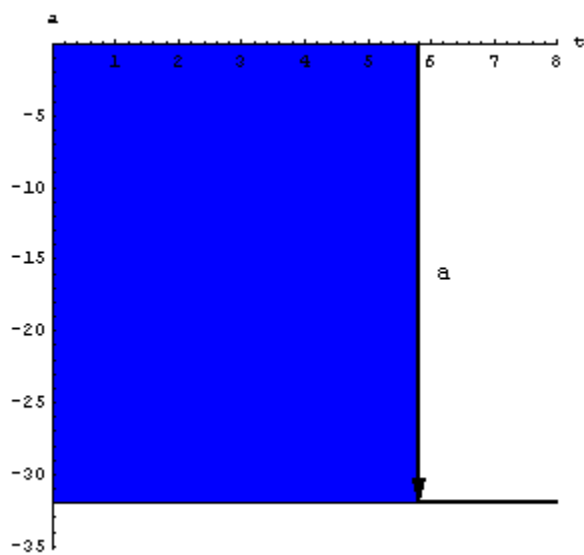
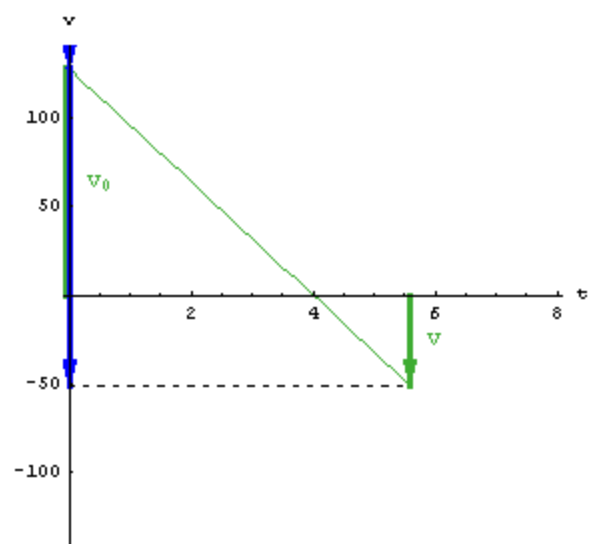
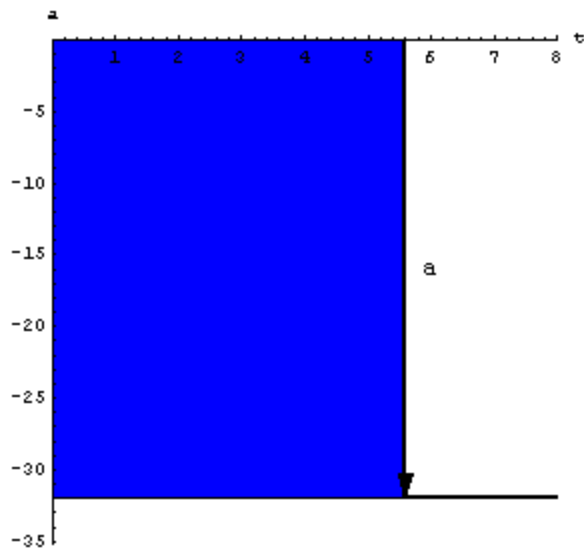


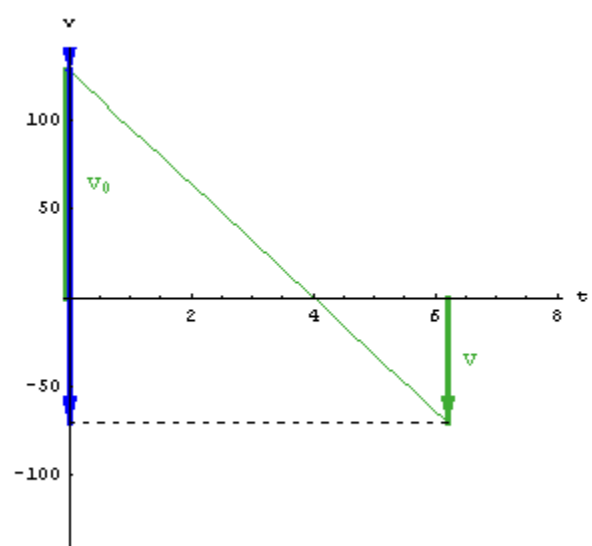
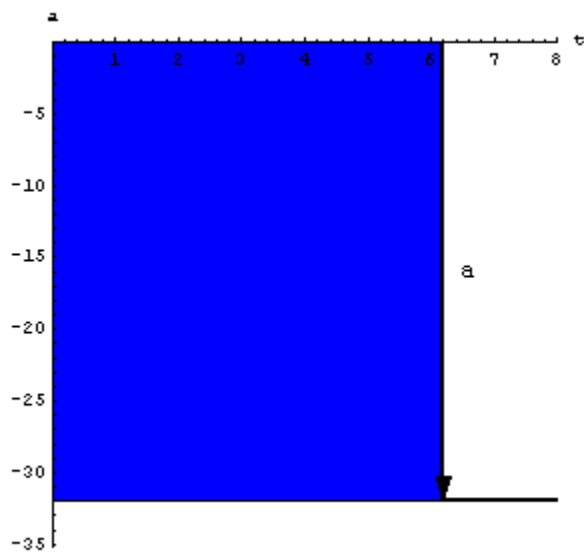
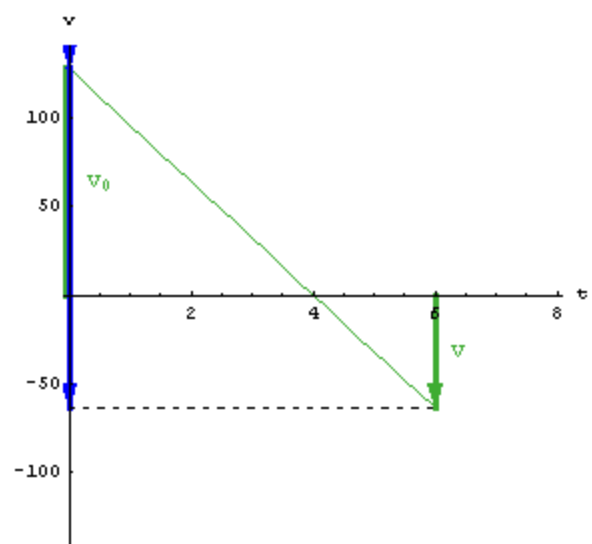
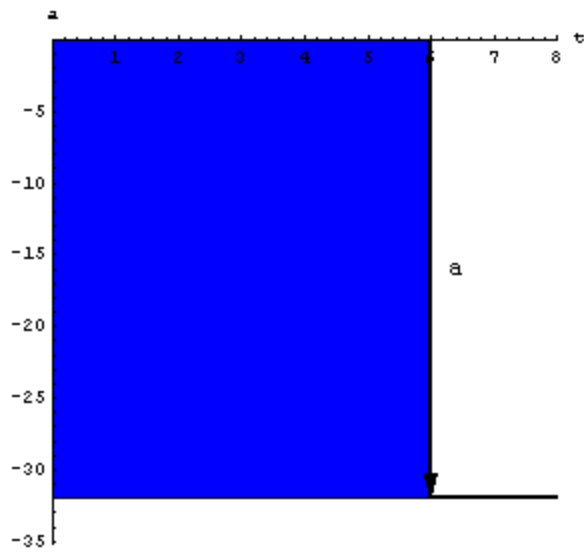


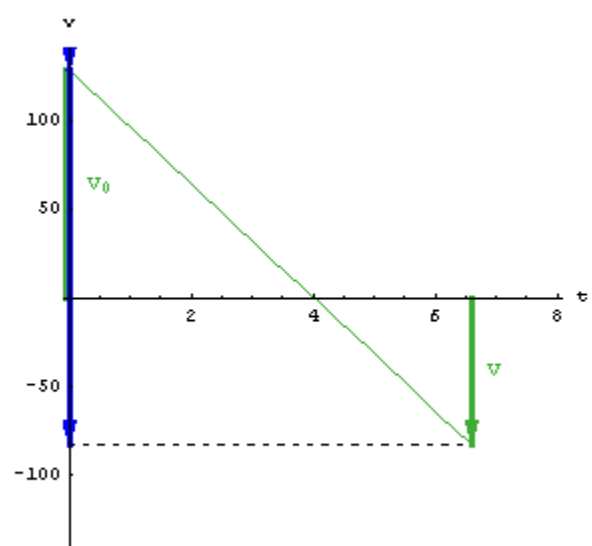
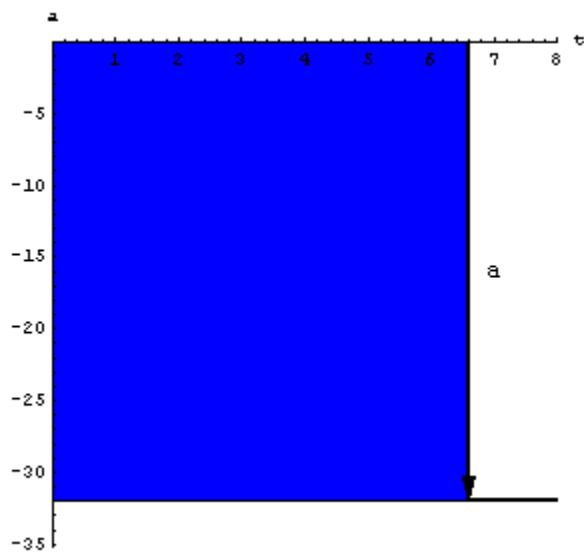
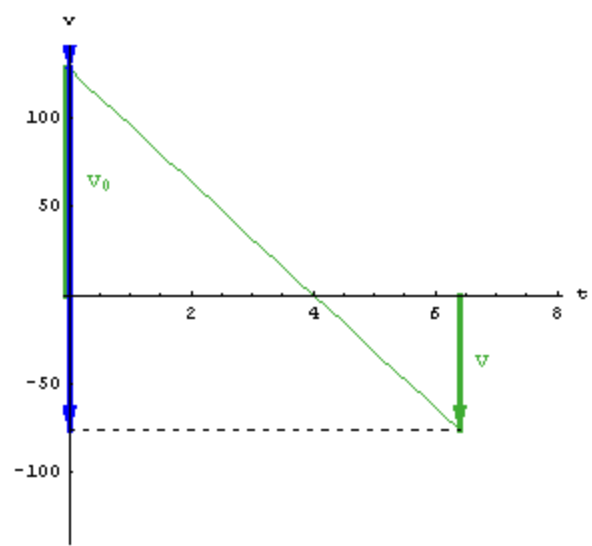
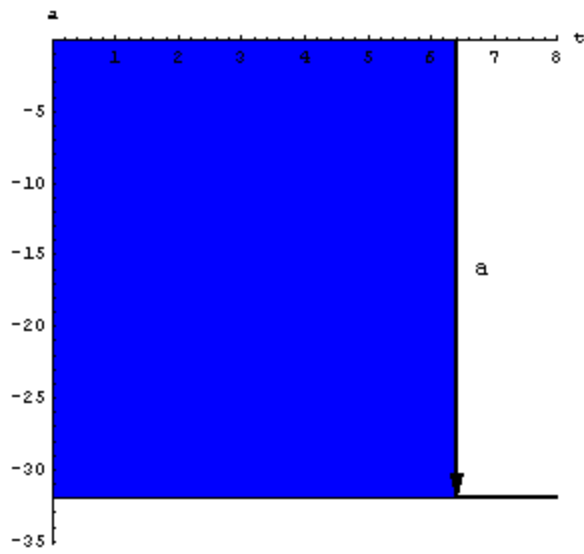


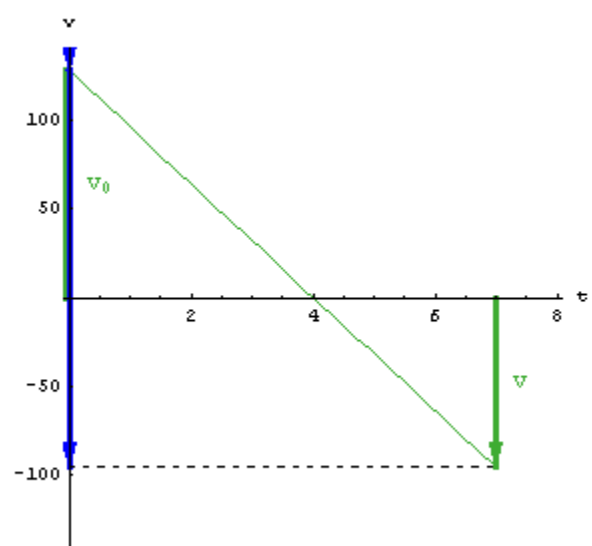
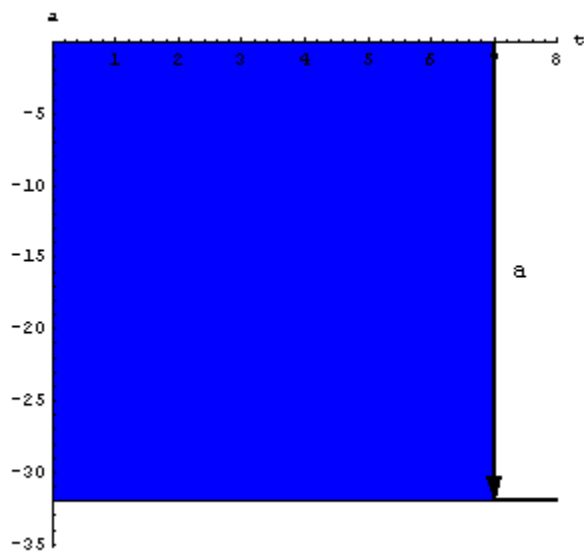
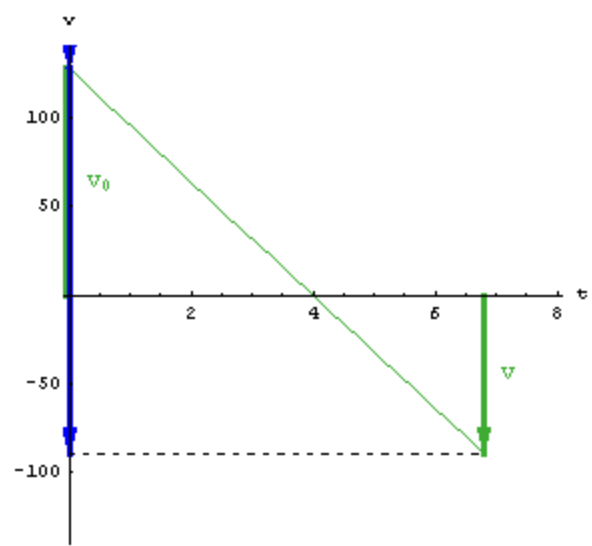
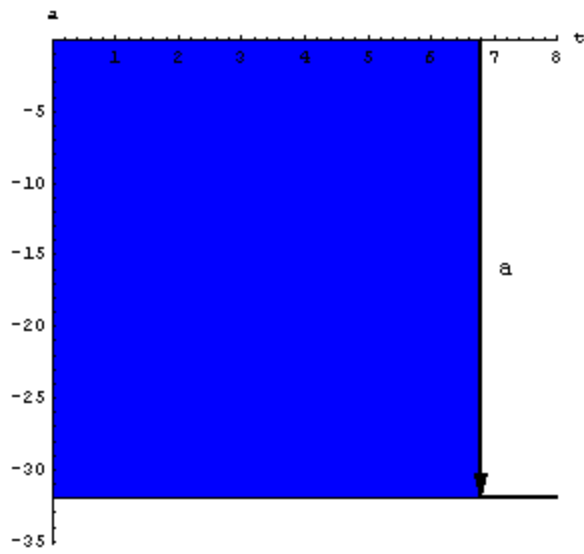


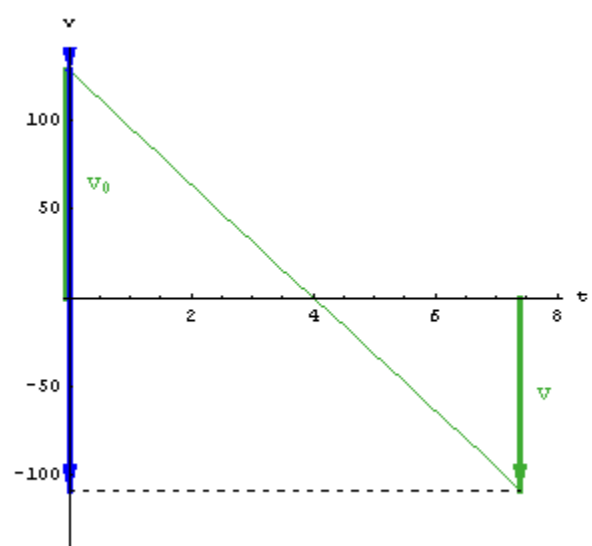
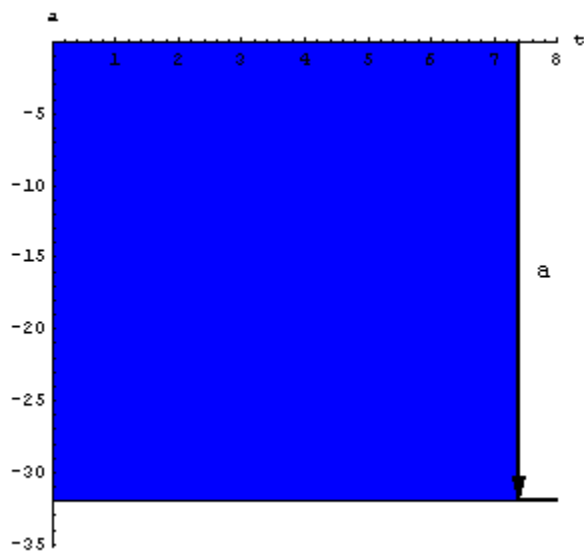
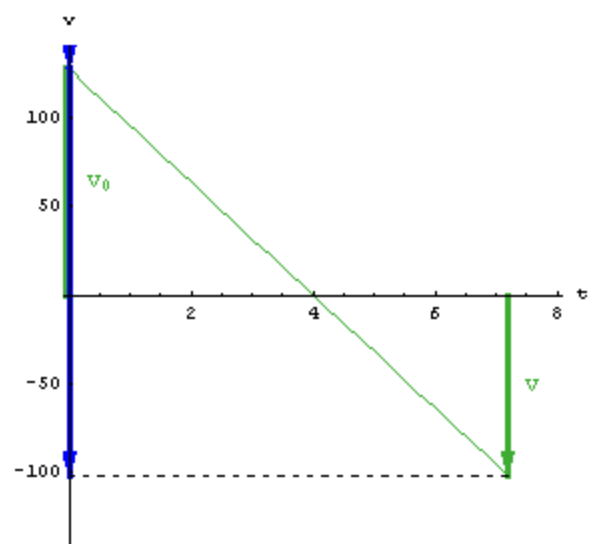
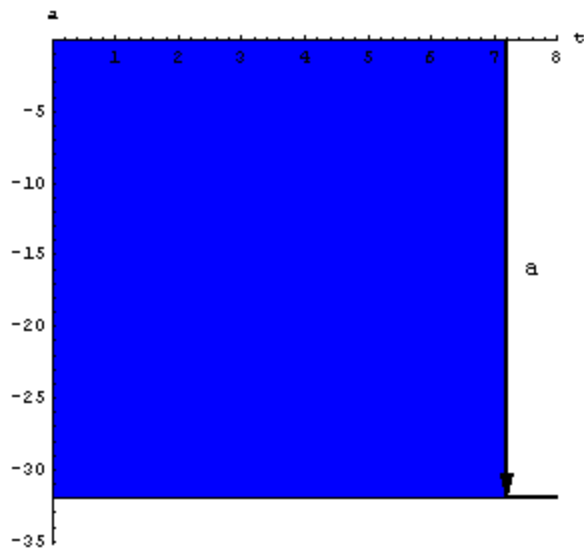


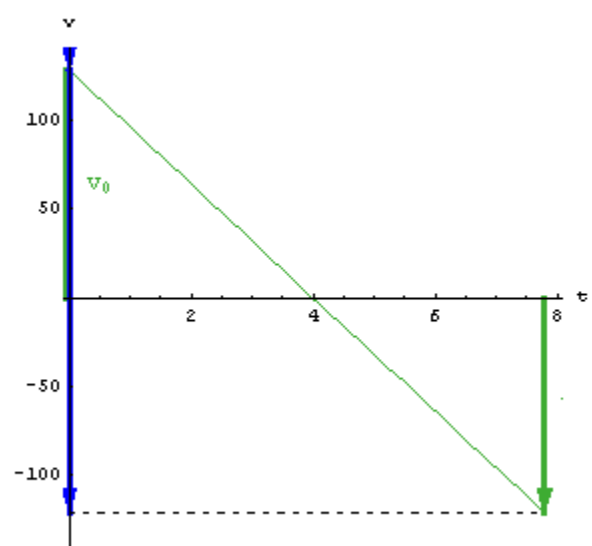
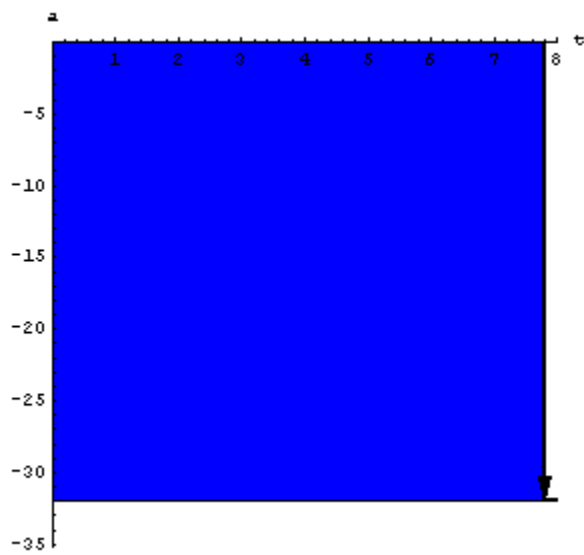
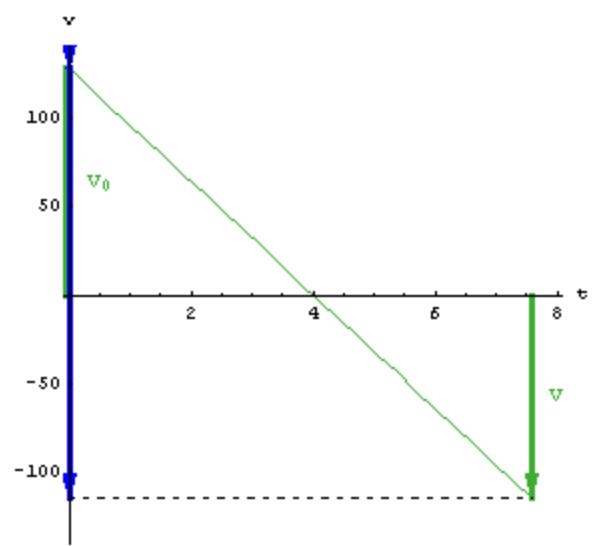
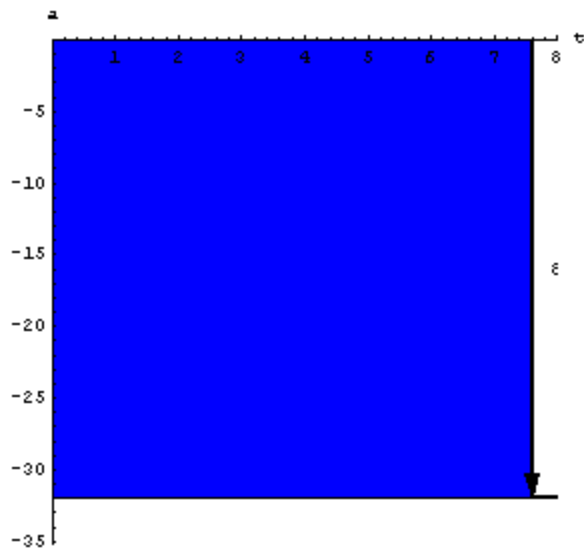


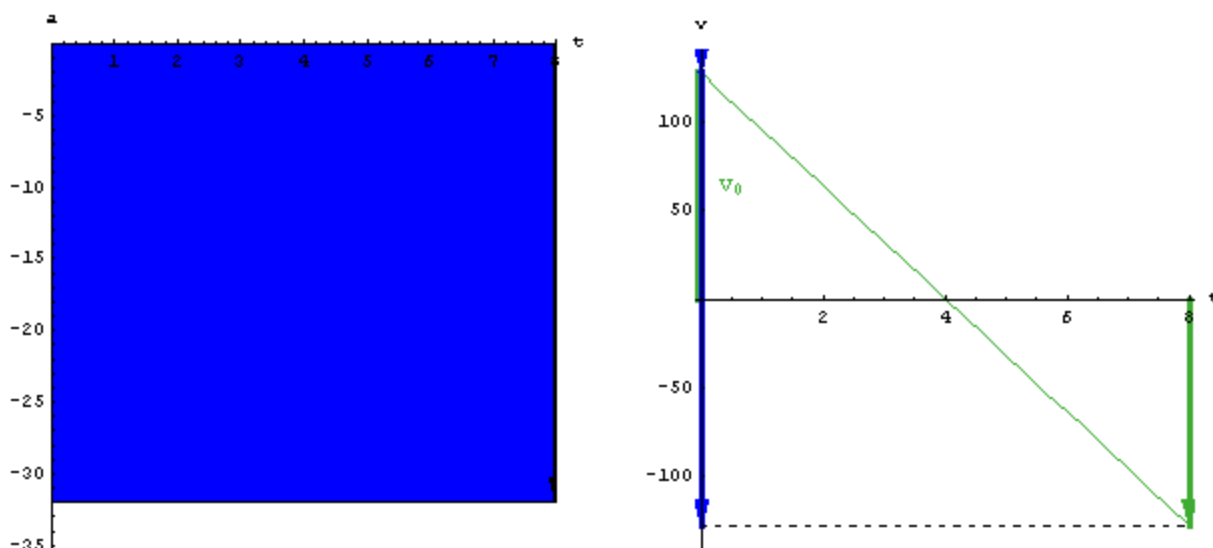












You Try It: Oscillation of a Mass and Spring

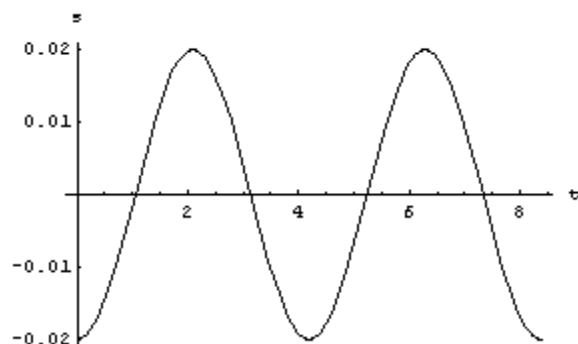
If you hang a mass from a spring, it will eventually come to rest in an equilibrium position where the weight of the mass is balanced by the tension in the spring. Now, if you pull the mass down slightly and let it go, it will move up and down or oscillate about the equilibrium position. If there were no air resistance and no friction in the mass and spring, it would oscillate forever with the same amplitude and frequency. Since the motion is periodic, the periodic trig functions are used to describe the motion. Suppose you were to pull the mass down 2 centimeters ($=0.02$ meters) below its equilibrium position and let it go. The position of the mass would be described with the following function.

In[43]:=

```
Clear[s];
```

```
s[t_] = -0.02 * Cos[1.5 * t];
```

```
Plot[s[t], {t, 0, 8 * Pi / 3}, AxesLabel -> {"t",
```



The time coefficient inside the cosine function is called the angular frequency of the oscillation. The period of the oscillation, that is, the time to go up and come back down to the release position, can be calculated from the angular frequency using the formula, $T = \frac{2\pi}{\omega}$, where T is the period in seconds and ω is the angular frequency in radians per second. In our example, the angular frequency of the oscillation is 1.5 radians per second. This value is determined by the size of the mass and the stiffness of the spring. In fact, the angular frequency of oscillation is equal to $\sqrt{k/m}$ where k is the stiffness of the spring in newtons per meter, for example, and m is the size of the mass, in kilograms. For the position function above, $\omega = \sqrt{k/m} = 1.5$ radians per second.

Determine the velocity and acceleration of the mass as it moves up and down, and graph them. Complete the *Mathematica* commands that follow by replacing the red "?" marks with the appropriate functions in *Mathematica* format (don't include the underscore "_" after the independent variable). The purple left cell bracket will change to black when you replace the question mark.

```
Clear[v, a];

v[t_] = D[?, t]

Plot[v[t], {t, 0, 8 * Pi / 3}, AxesLabel -> {"t",

a[t_] = D[?, t]

Plot[a[t], {t, 0, 8 * Pi / 3}, AxesLabel -> {"t",
```

Let's complete the circle of ideas. Now that you know the acceleration function and an initial condition, fill in the following commands by replacing the red entries with the appropriate function and initial condition to reconstruct the velocity function, $v(t)$. Recall that you release the mass from rest after pulling it down 0.02 meters. You will need to replace t , the independent variable inside the integrand function, with u so that the **Integrate[]** command

will work correctly.

```
Clear[v, v0];

v0 = initialvelocity;

v[t_] = (Integrate[ acceleration, {u, 0, t}] + v0) / m;

Plot[v[t], {t, 0, 8 * Pi / 3}, AxesLabel → {"t",
```

Next, use the special **velocity[]** function to see the visual reconstruction of $v(t)$.

Note: The next command generates a lot of graphics, which may fill up your computer's memory. Before executing the next command, pull down the Kernel menu and select Delete All Output. All computed values are saved when you do this.

```
velocity[a[t], {t, 0.0, 8.0 * Pi / 3.0}, v0];
```

Now you can reconstruct the position function, $s(t)$. Fill in the following commands by replacing the red entries appropriately. Recall that the mass is released from a position 0.02 meters below the equilibrium position.

```
Clear[s, s0];

s0 = initialposition;

s[t_] = (Integrate[velocity, {u, 0, t}] + s0) / m;

Plot[s[t], {t, 0, 8 * Pi / 3}, AxesLabel → {"t",
```

Use **position[]** to see the visual reconstruction of $s(t)$.

Note: The next command generates a lot of graphics, which may fill up your computer's memory. Before executing the next command, pull down the Kernel menu and select Delete All Output. All computed values are saved when you do this.

```
position[v[t], {t, 0, 8 * Pi / 3}, s0];
```

We can make some observations about this motion. The first is that the acceleration is in the direction opposite the position. When the mass is below the equilibrium position, the stretch in

the spring increases, thus exerting a net upward force on the mass, accelerating it in that direction. When the mass is above the equilibrium position, the stretch in the spring is reduced resulting in a net downward force on the mass, accelerating it in that direction. The second observation is that the acceleration is proportional to the position. The first two observations are demonstrated mathematically by noting that

$$\text{if } s(t) = -A \cos(\omega t), \text{ then } a(t) = \frac{d^2 s}{dt^2} = A\omega^2 \cos(\omega t) = \omega^2 (A \cos(\omega t)) = -\omega^2 s(t),$$

where ω and A are constants.

Another thing to note is that the velocity and acceleration are 90 degrees or $\frac{\pi}{2}$ radians out of phase with one another. During the first quarter-cycle of the motion, the velocity and acceleration are both positive and the mass is speeding up, moving upward. During the second-quarter cycle, the velocity is positive and the acceleration is negative, and the mass is still moving upward but now is slowing down. During the third quarter-cycle, the velocity and acceleration are both negative, and the mass is speeding up as it moves downward. During the fourth quarter-cycle, the velocity is negative and the acceleration is positive, and the mass is slowing down as it moves downward. Overlaying the graphs of the velocity and acceleration functions shows these relationships.

```
Plot[{s[t], v[t], a[t]}, {t, 0, 4 * Pi / 3}, Axes
PlotStyle -> {{RGBColor[1, 0, 0]}, {RGBColor[1,
{RGBColor[0, 0, 1]}}];
```

The velocity is $v(t) = -A\omega \sin(\omega t)$ and the acceleration is $a(t) = -A\omega^2 \cos(\omega t)$. From basic trigonometry, we know that the sine and cosine functions are 90 degrees or $\frac{\pi}{2}$ radians out of phase with one another.

The final observation is that at the instant that the mass passes the equilibrium position, the velocity reaches its extreme values (positive on the way up and negative on the way down): its speed is at its maximum value, and its acceleration is zero.

Part III: Decaying Oscillations

A more realistic model for oscillations accounts for the loss of energy that results in a decay in the amplitude of the oscillations. In mechanical systems, this energy loss is usually due to friction (objects rubbing against one another in some way). For the motion of the mass hanging from a spring, a more realistic position function has amplitude that decays exponentially.

In[46]:=

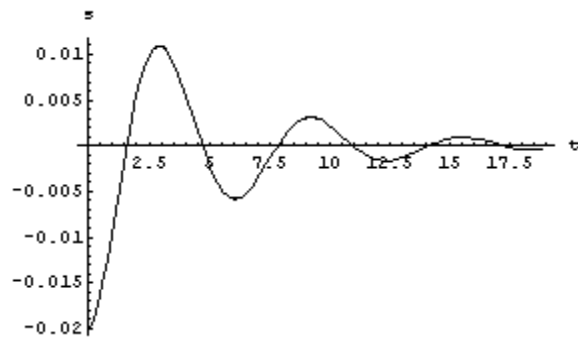
```

Clear[s];

s[t_] = -0.02 * Exp[-t / 5] * Cos[t];

Plot[s[t], {t, 0, 6 * Pi}, PlotRange -> All,
  AxesLabel -> {"t", "s"}];

```



Let's find the velocity and acceleration, and graph them.

In[49]:=

```
v[t_] = D[s[t], t] // Simplify
```

Out[49]=

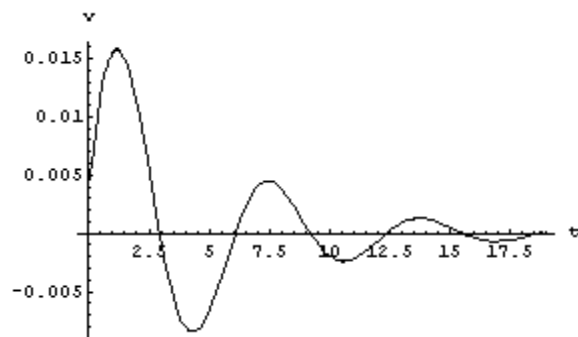
```
Exp[-t/5] (0.004 Cos[t] + 0.02 Sin[t])
```

In[50]:=

```

Plot[v[t], {t, 0, 6 * Pi}, PlotRange -> All,
  AxesLabel -> {"t", "v"}];

```



In[51]:=

```
a[t_] = D[v[t], t] // Simplify
```

```
Out[51]=
```

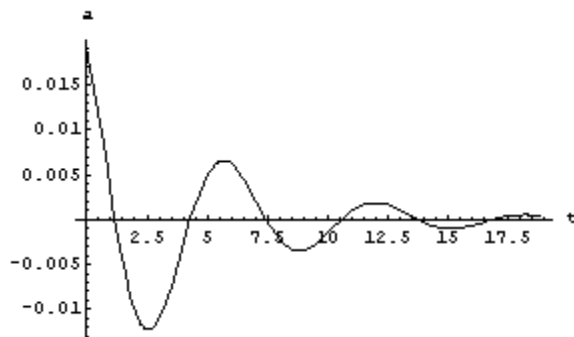
```

$$e^{-t/5} (0.0192 \cos[t] - 0.008 \sin[t])$$

```

```
In[52]:=
```

```
Plot[a[t], {t, 0, 6 * Pi}, PlotRange -> All,
  AxesLabel -> {"t", "a"}];
```



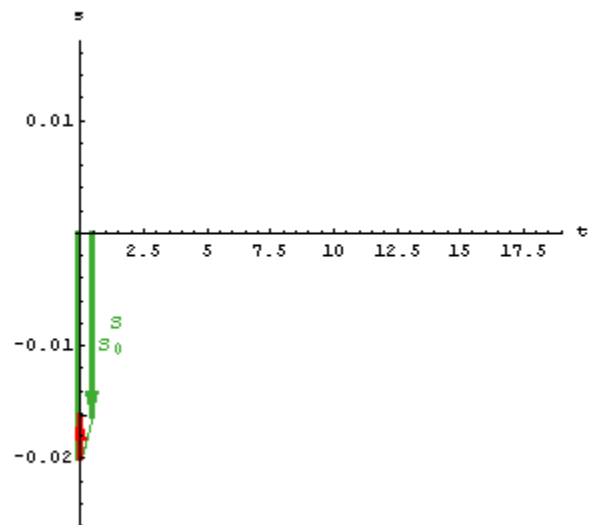
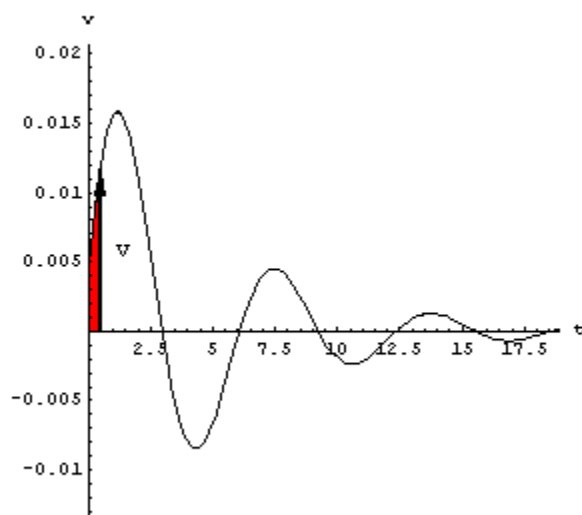
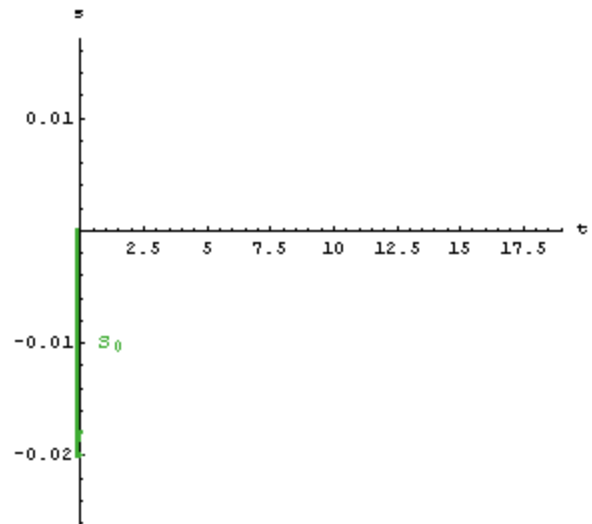
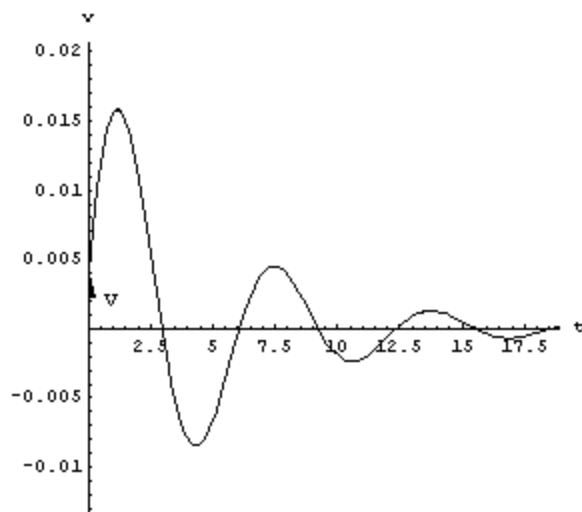
It is apparent that the amplitudes of the velocity and acceleration also decay with time.

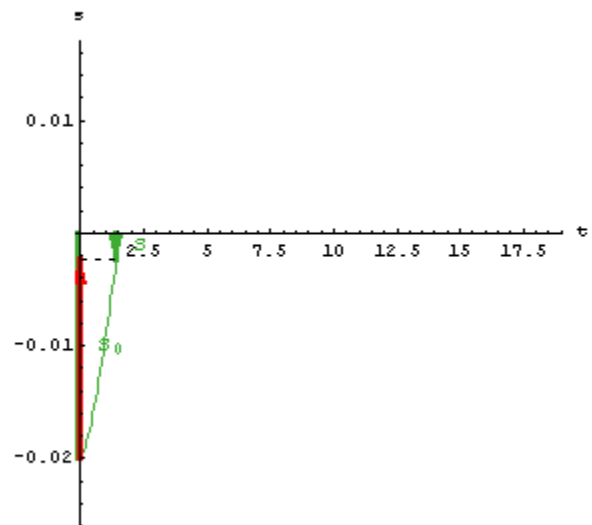
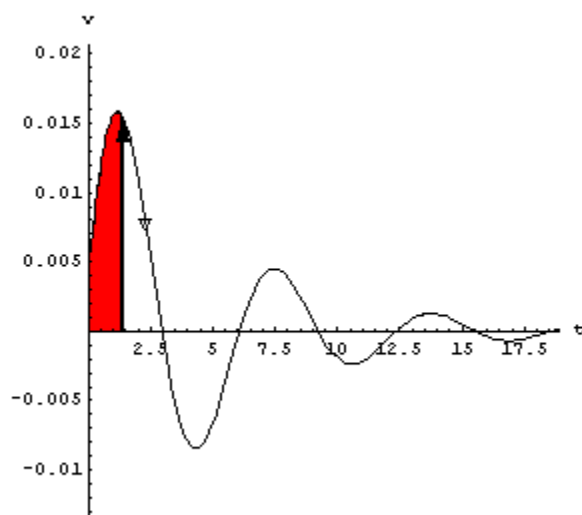
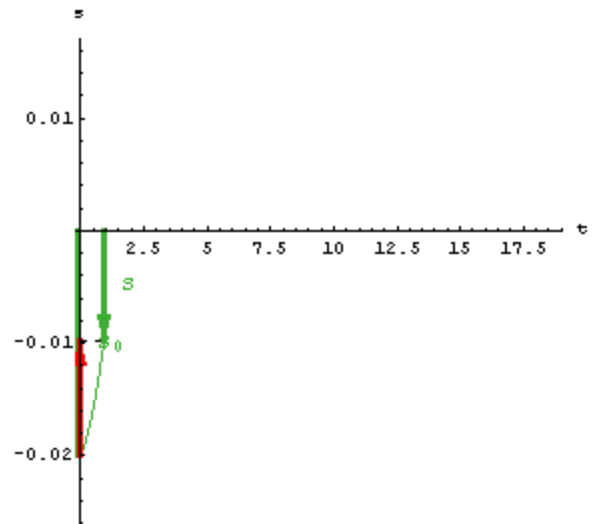
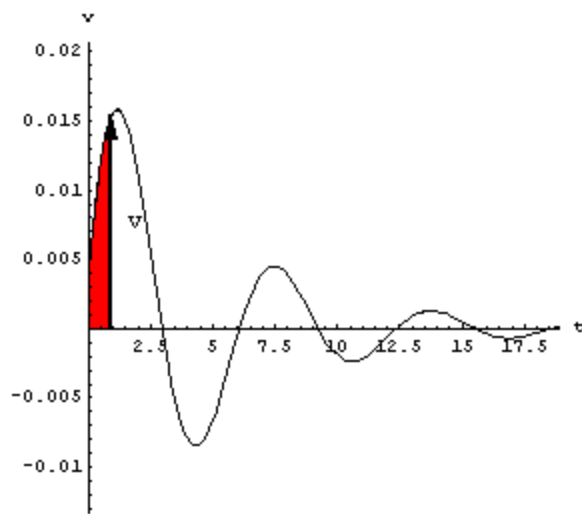
You can analyze the motion using the **position[]** and **velocity[]** commands.

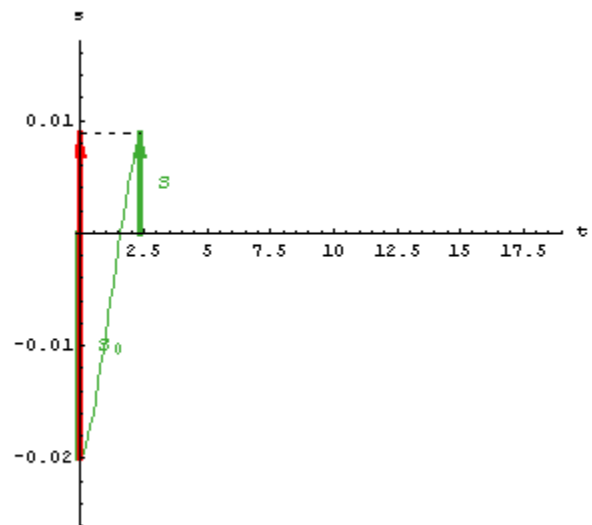
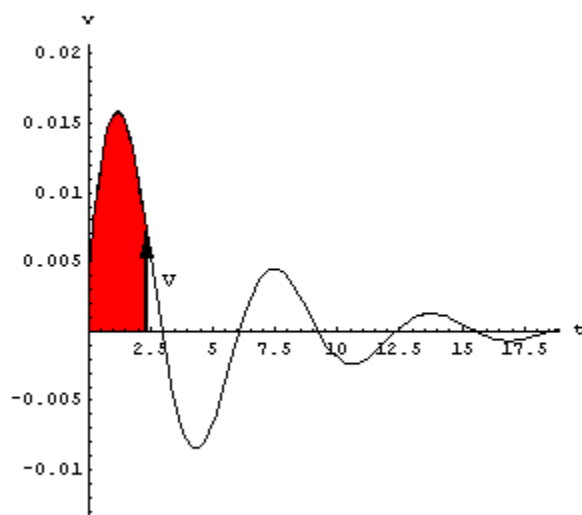
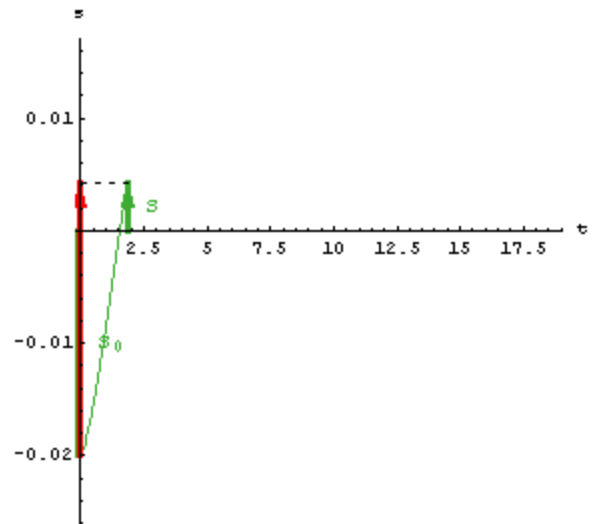
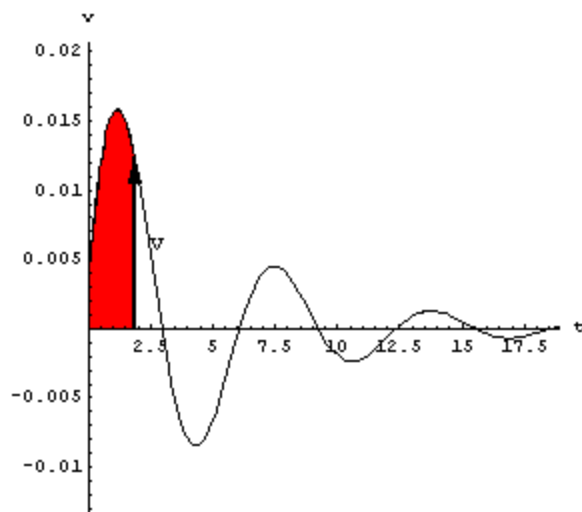
Note: The next two commands generate a lot of graphics, which may fill up your computer's memory. Before executing them, pull down the Kernel menu and select Delete All Output. All computed values are saved when you do this.

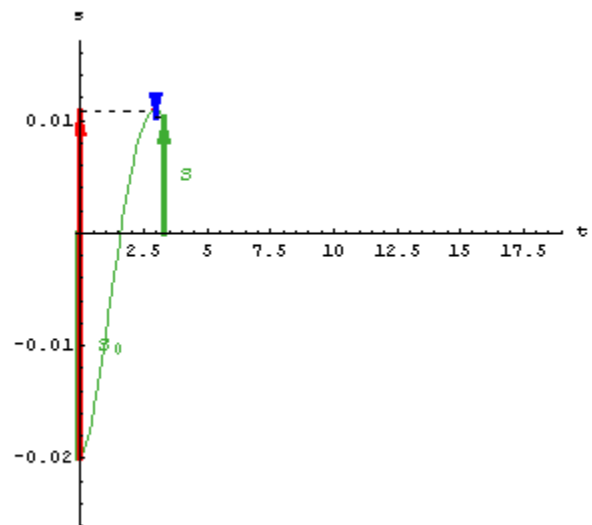
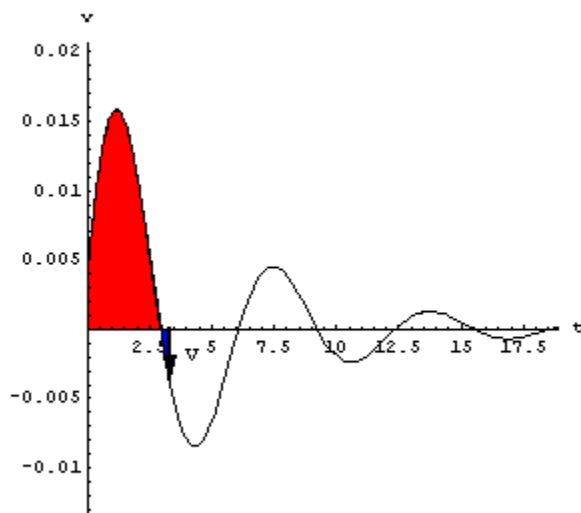
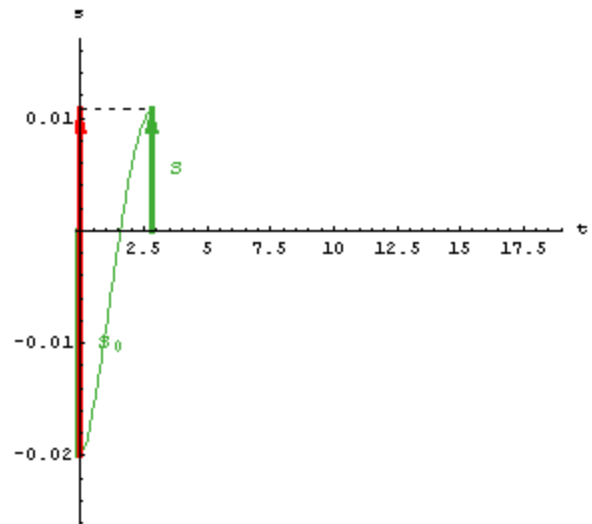
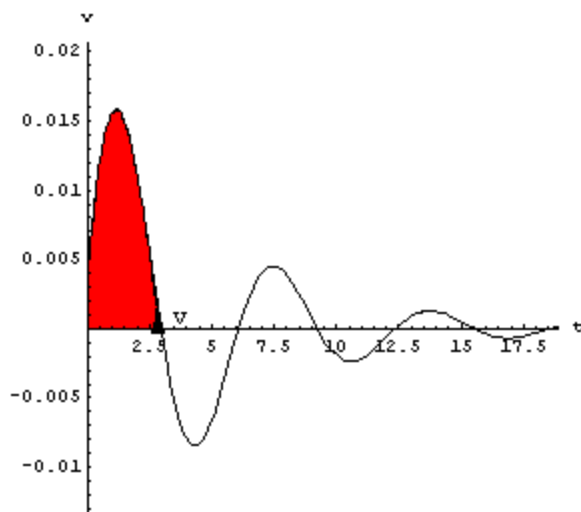
```
In[53]:=
```

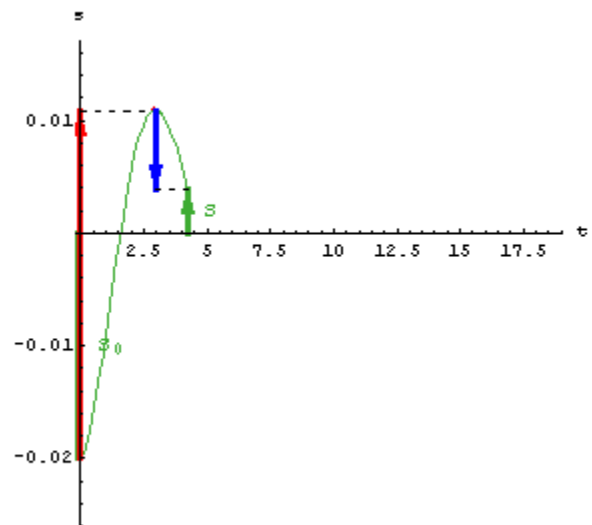
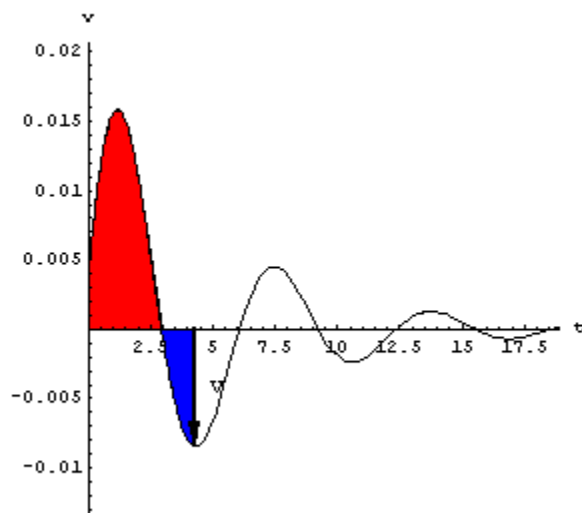
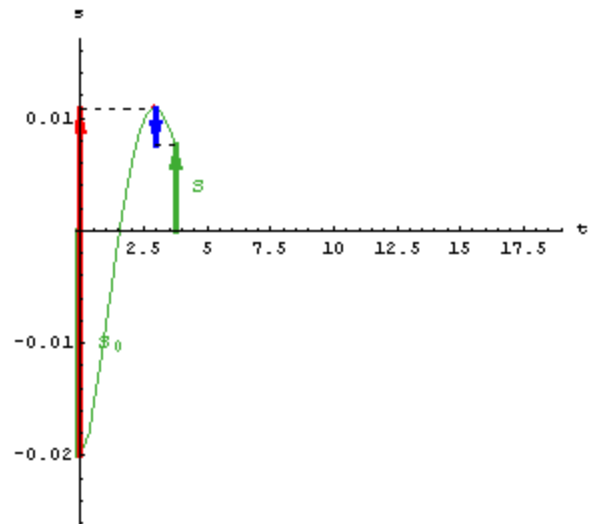
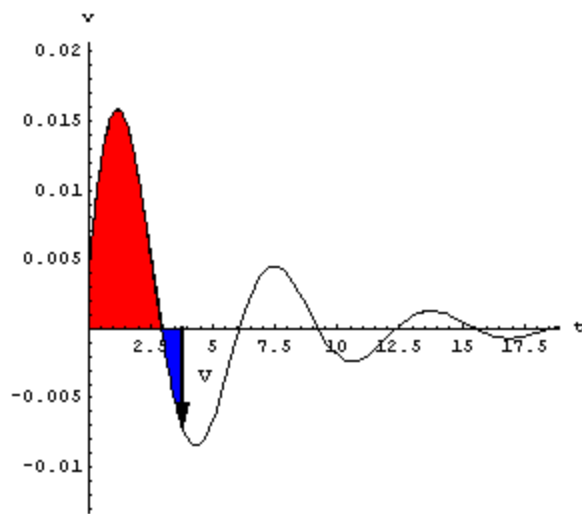
```
position[v[t], {t, 0, 6 * Pi}, s[0]]:
```

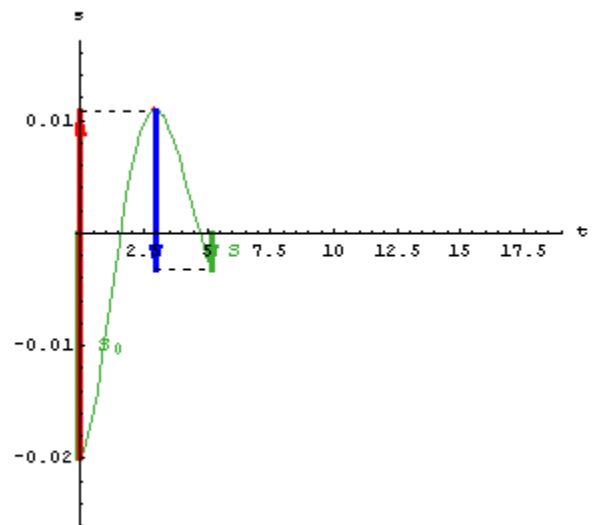
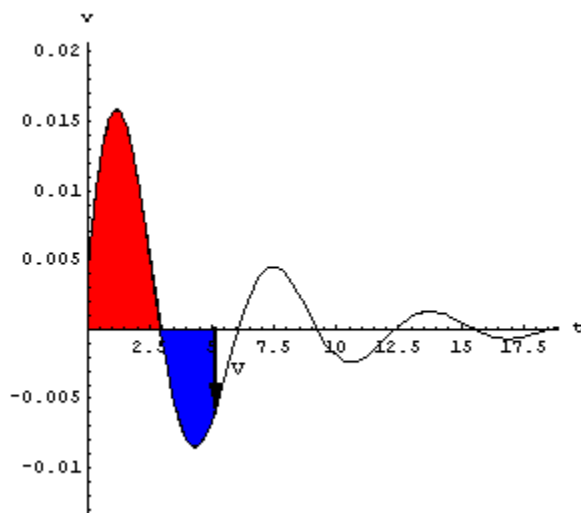
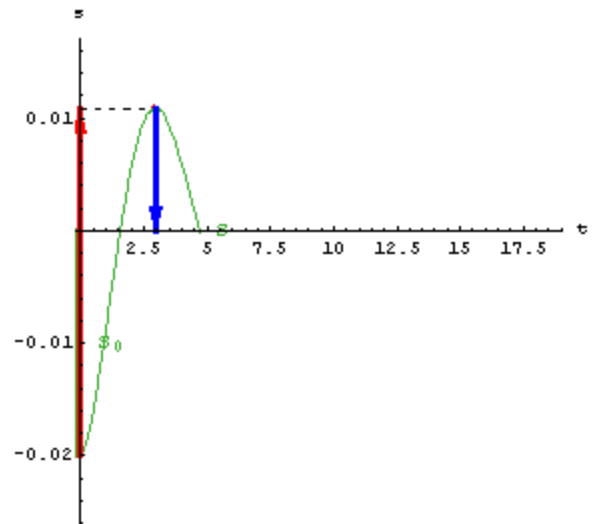
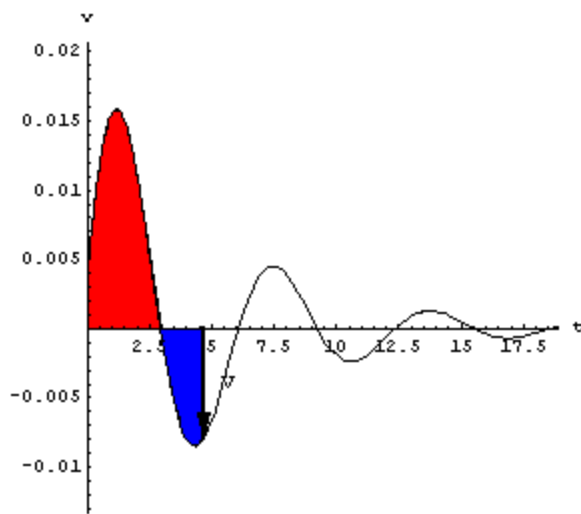


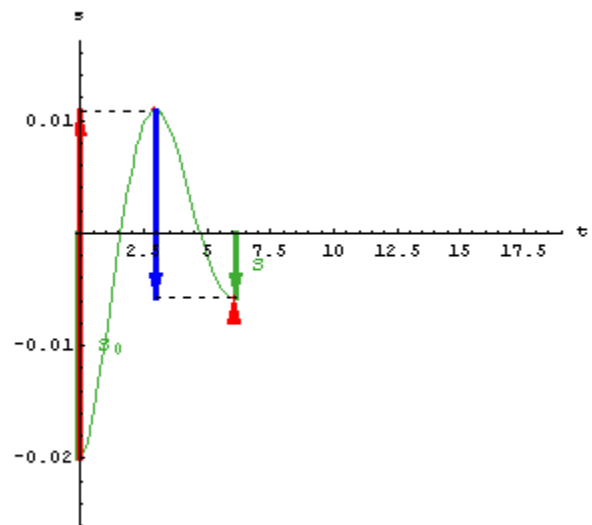
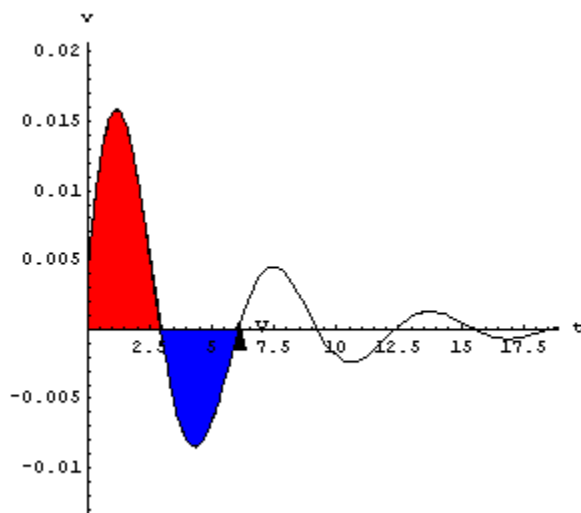
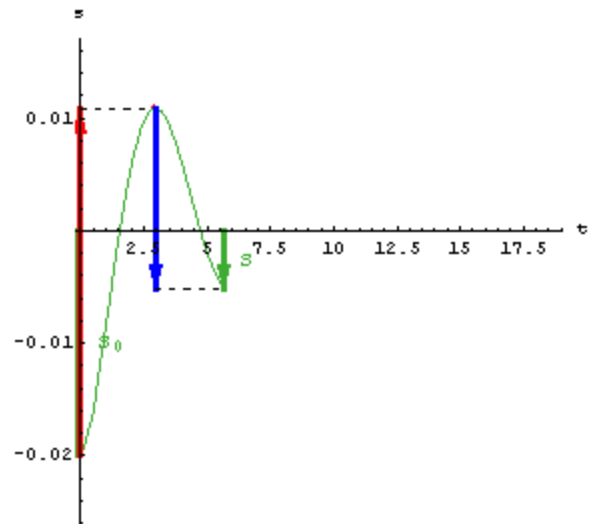
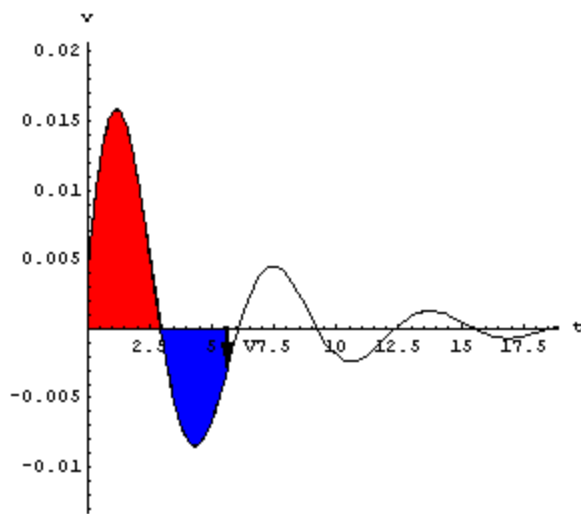


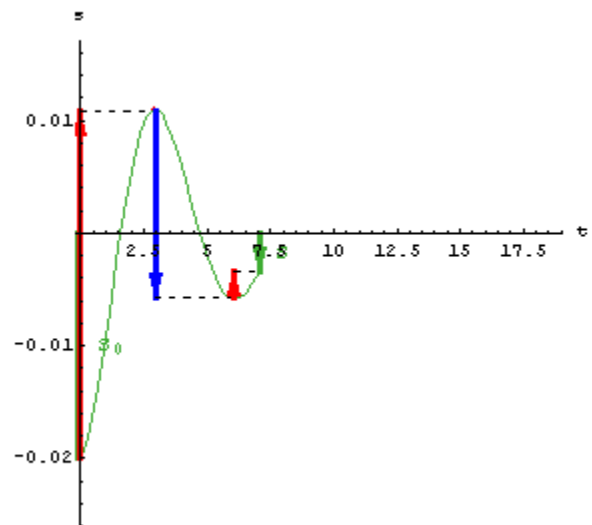
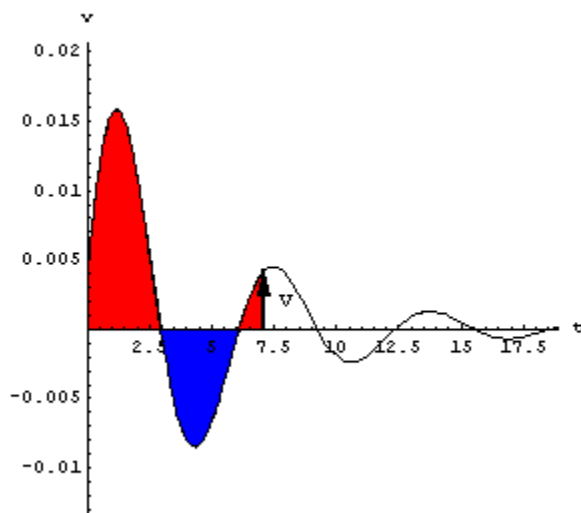
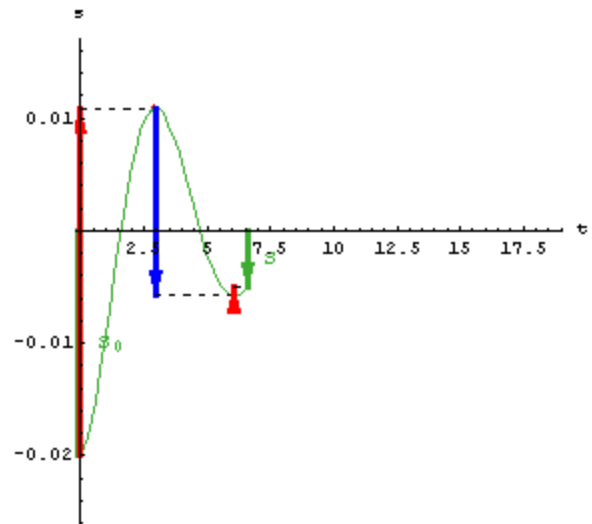
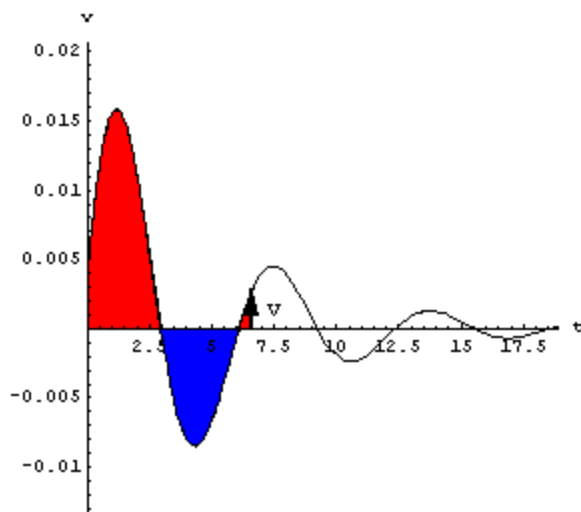


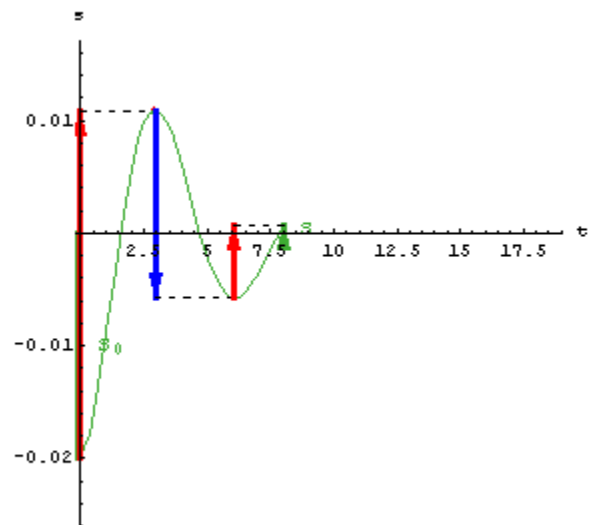
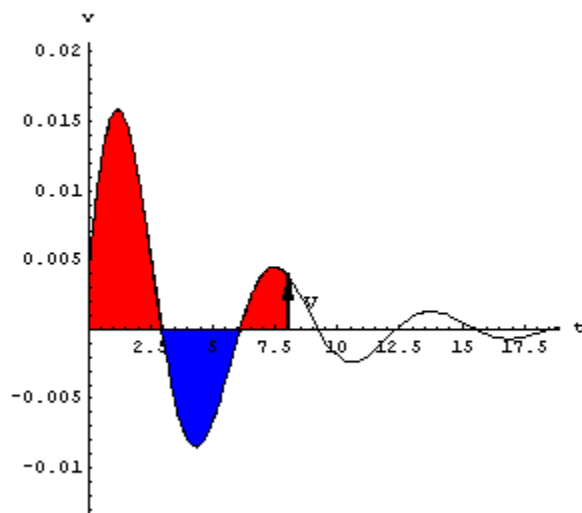
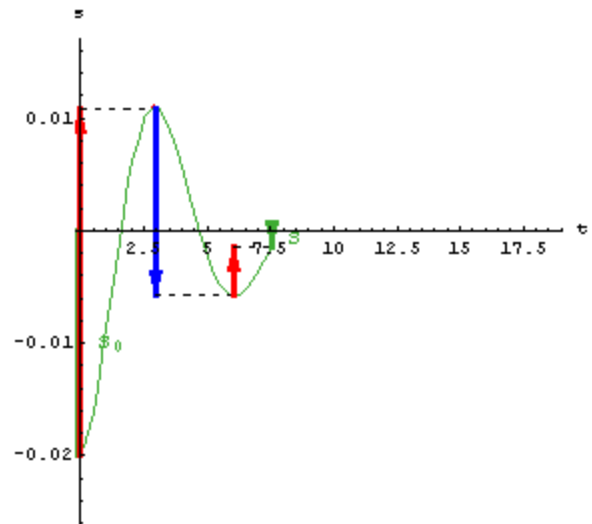
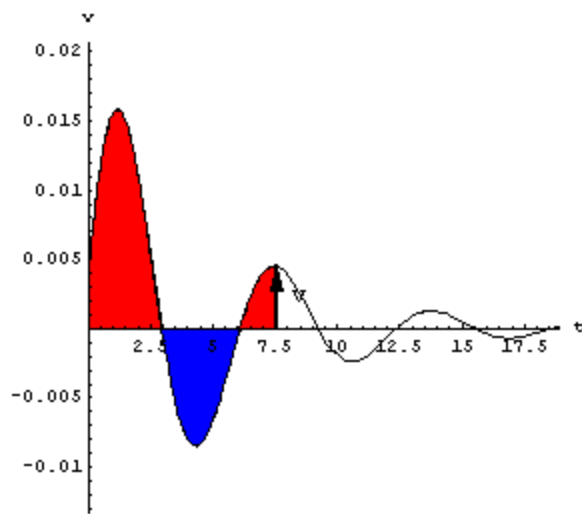


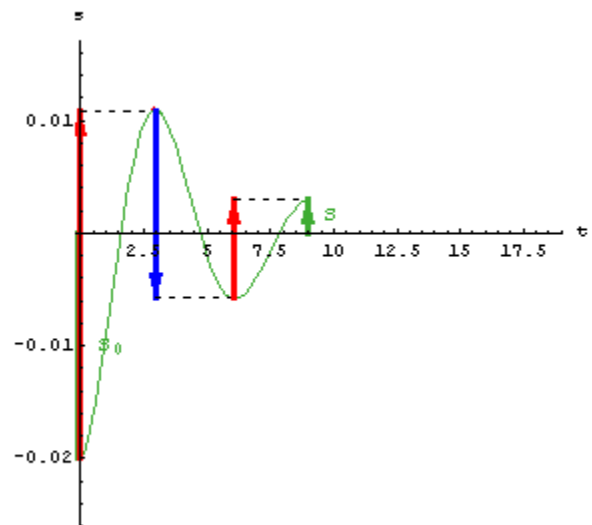
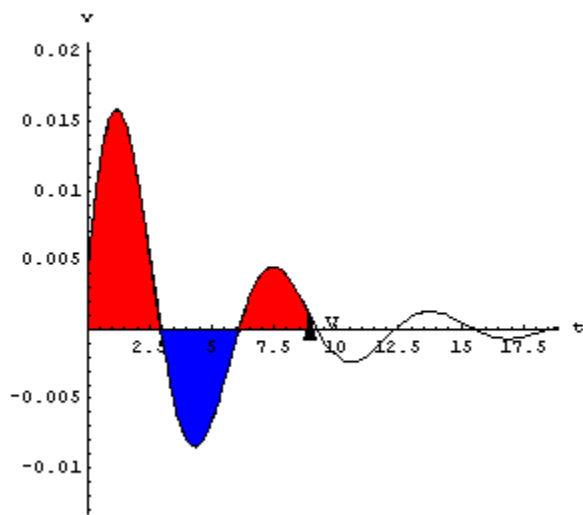
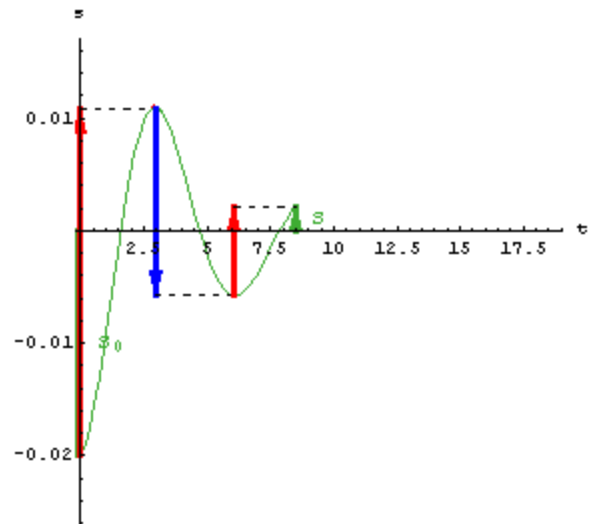
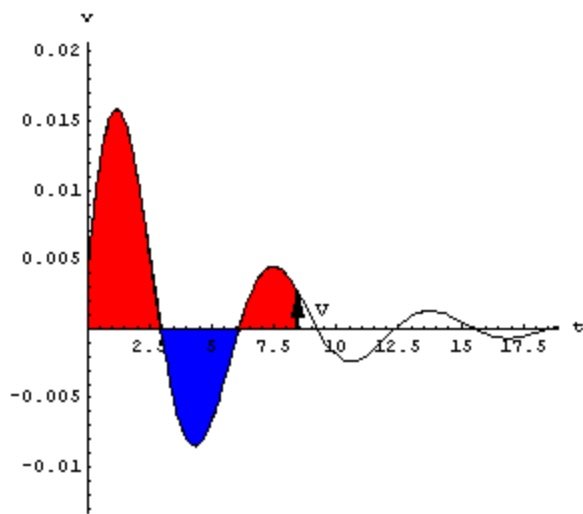


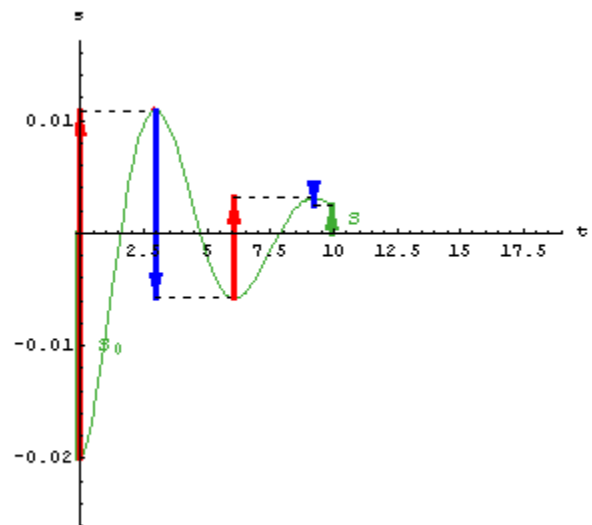
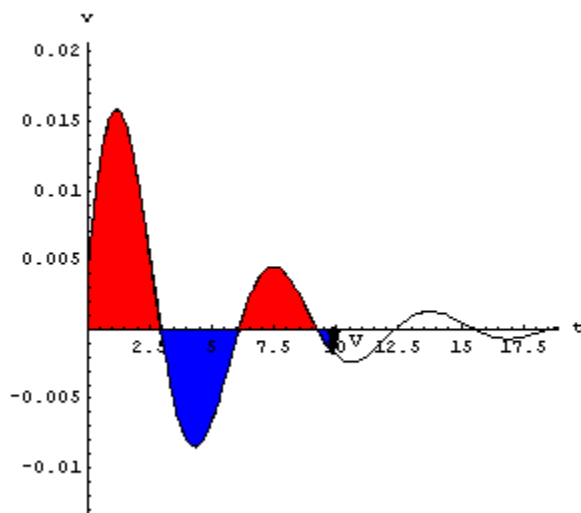
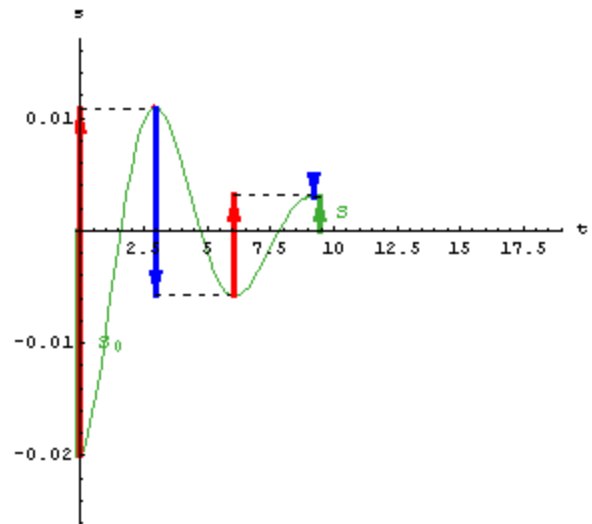
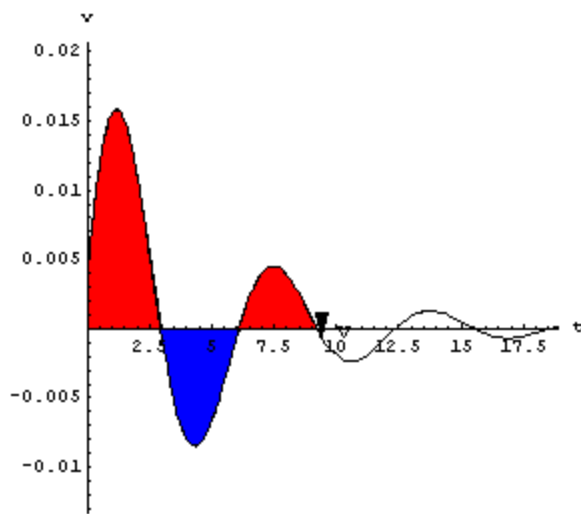


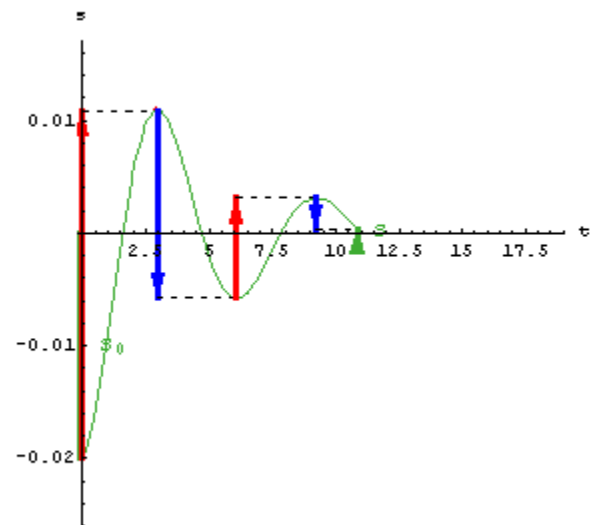
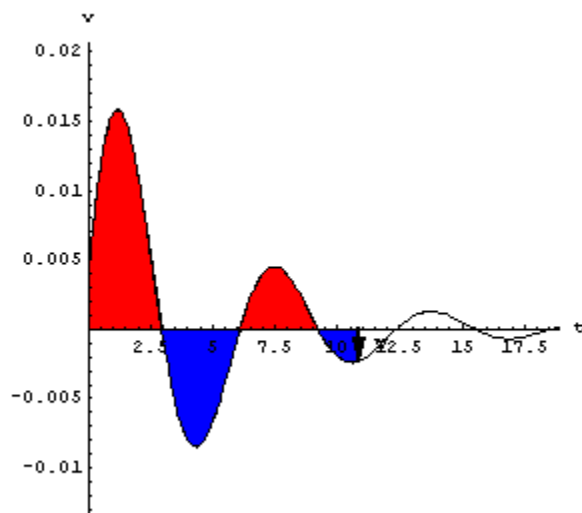
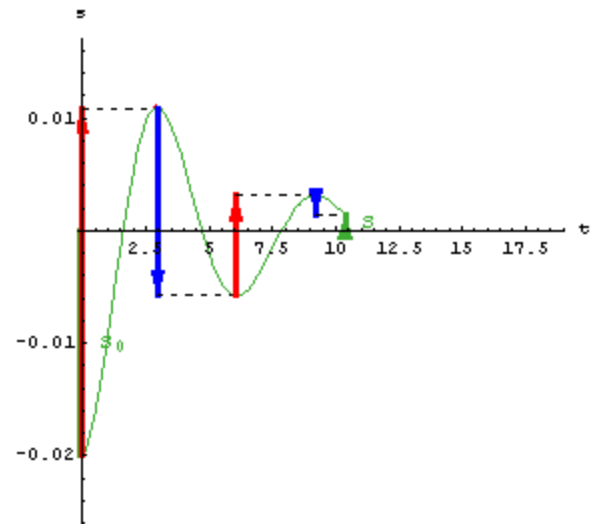
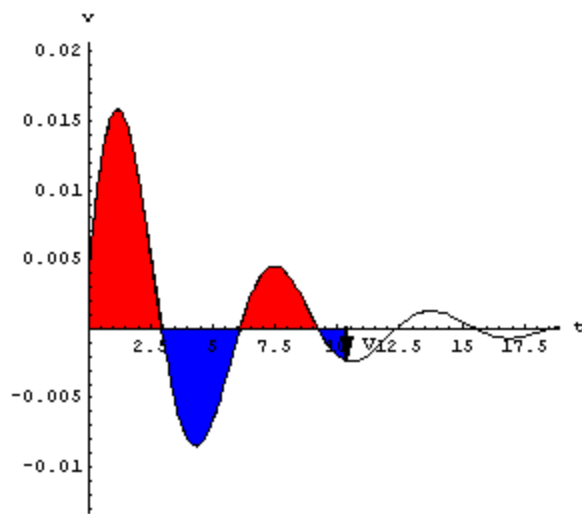


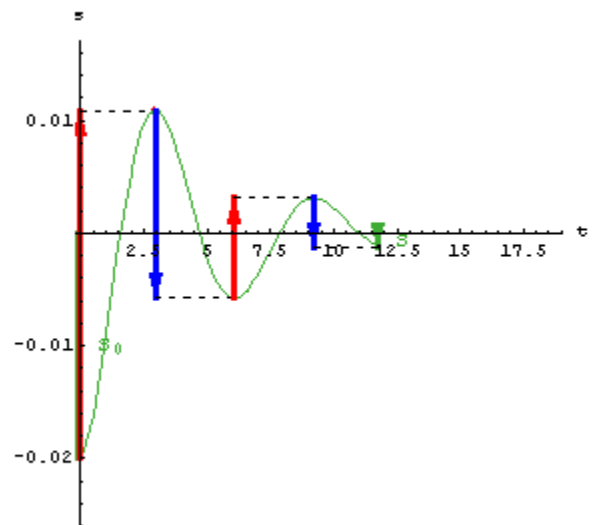
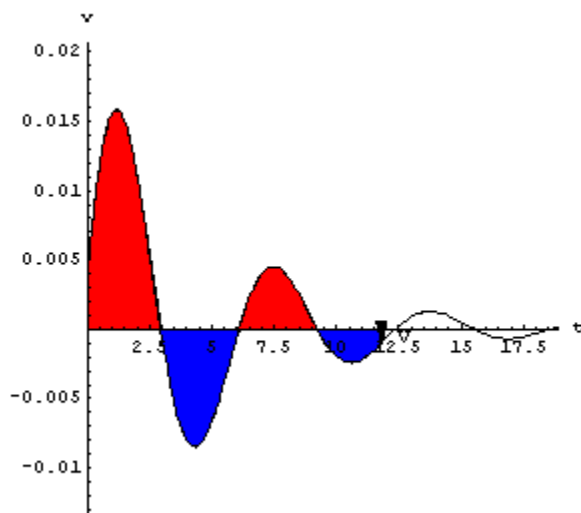
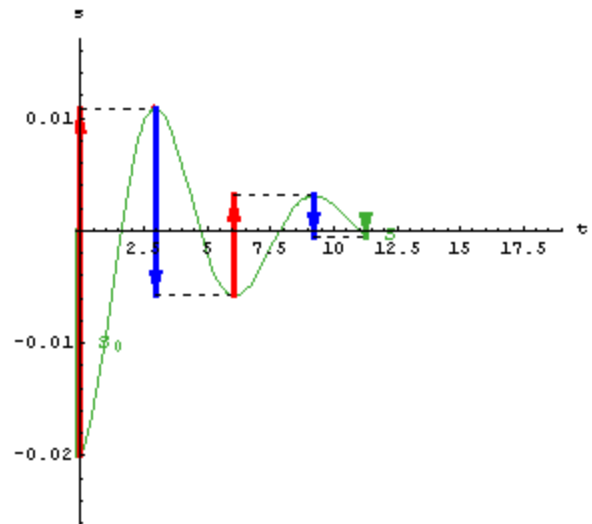
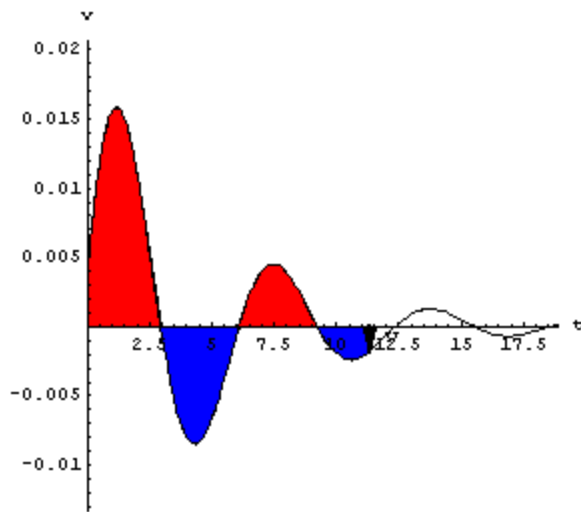


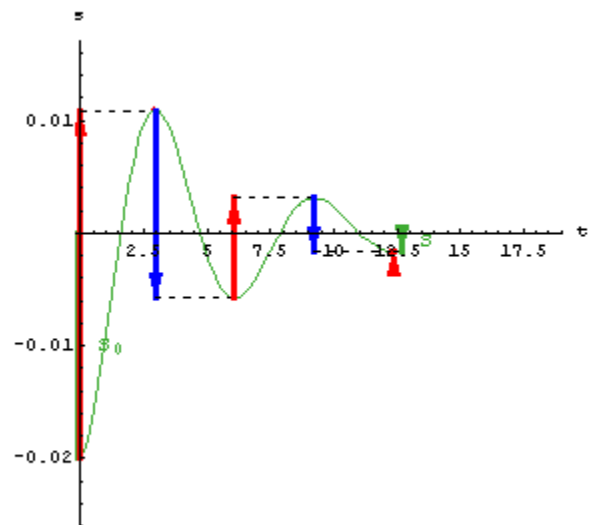
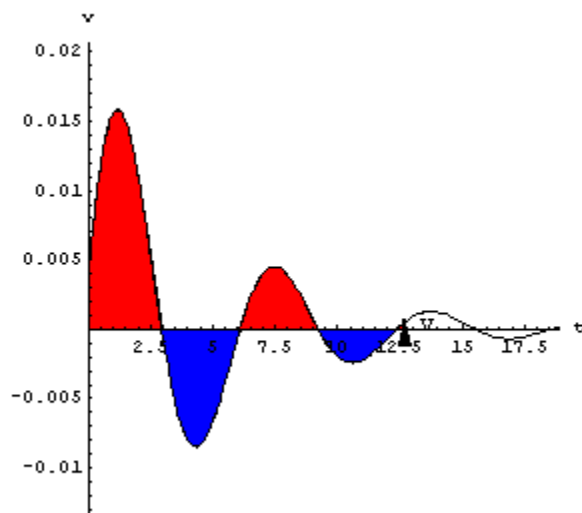
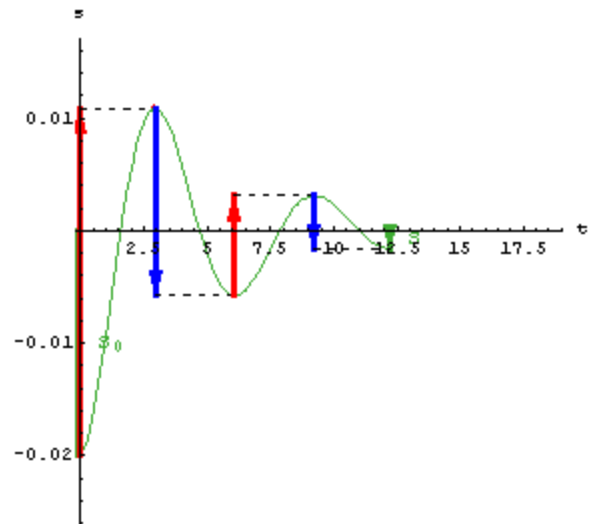
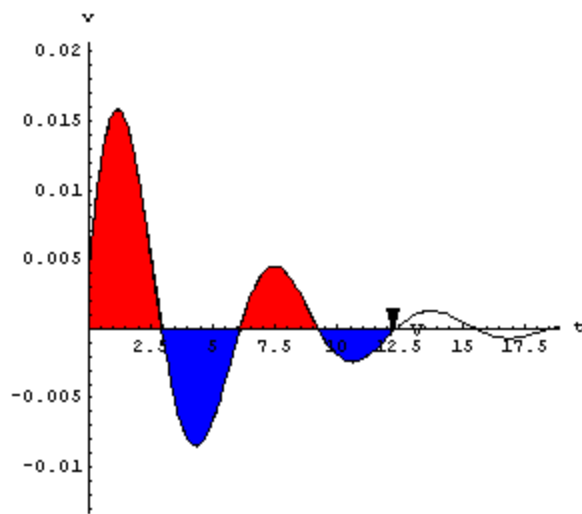


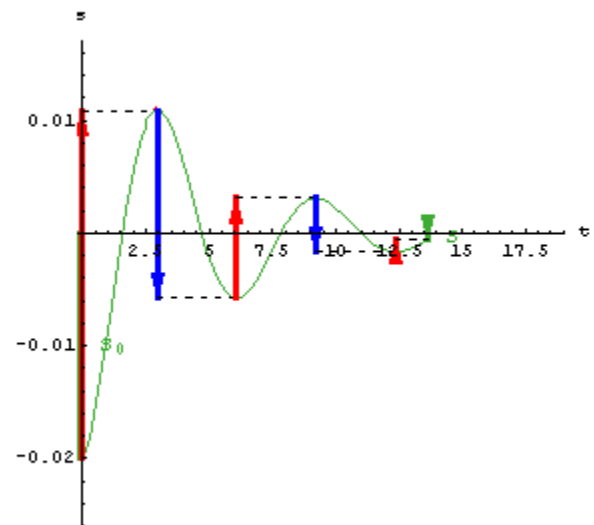
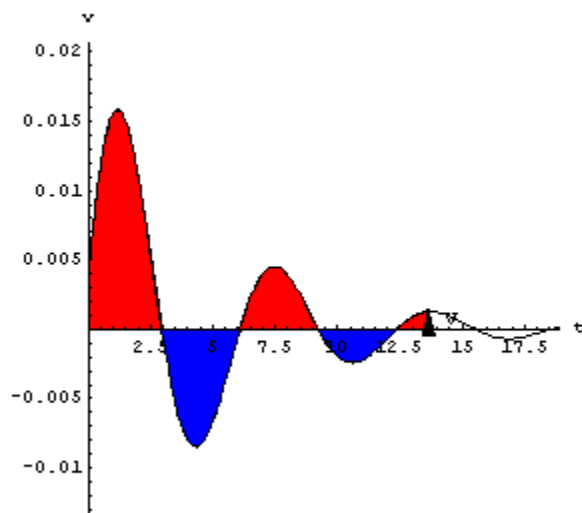
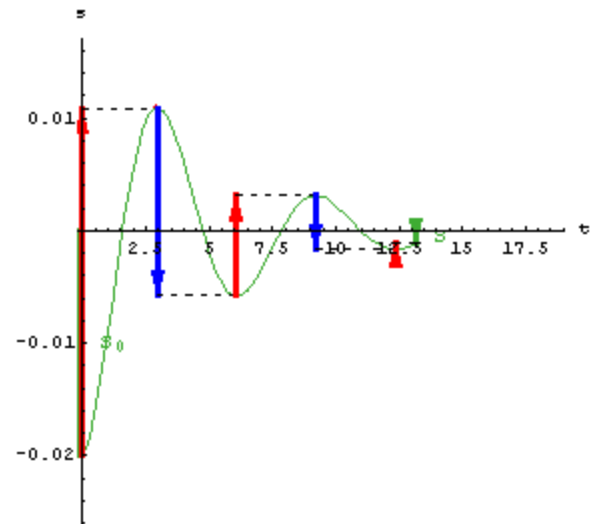
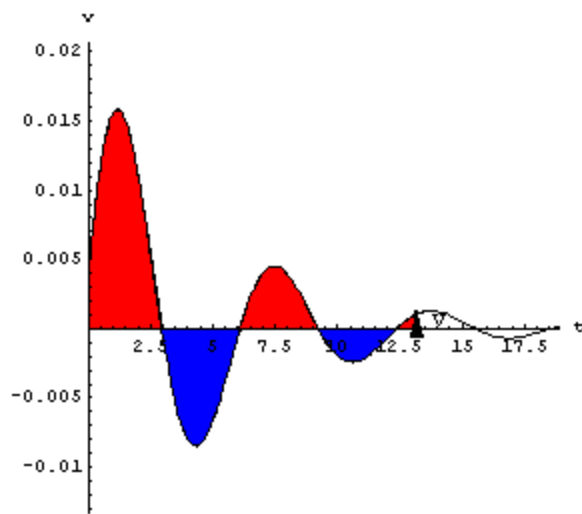


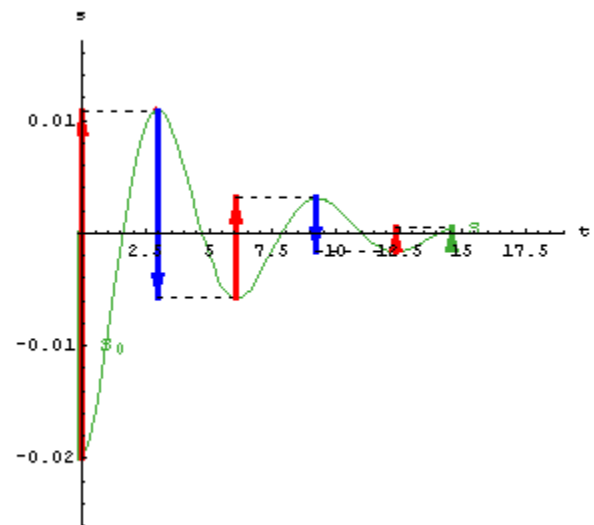
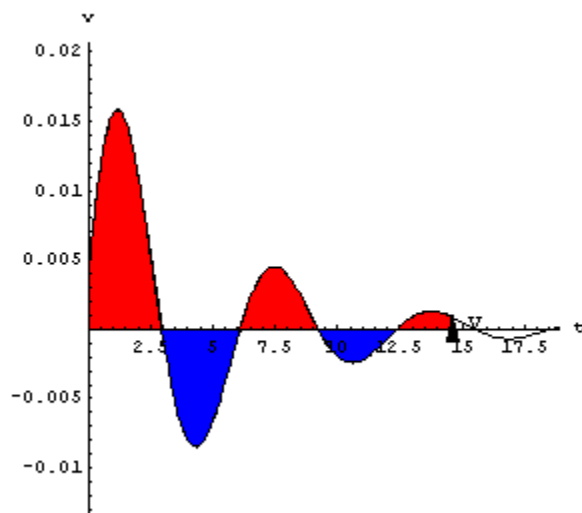
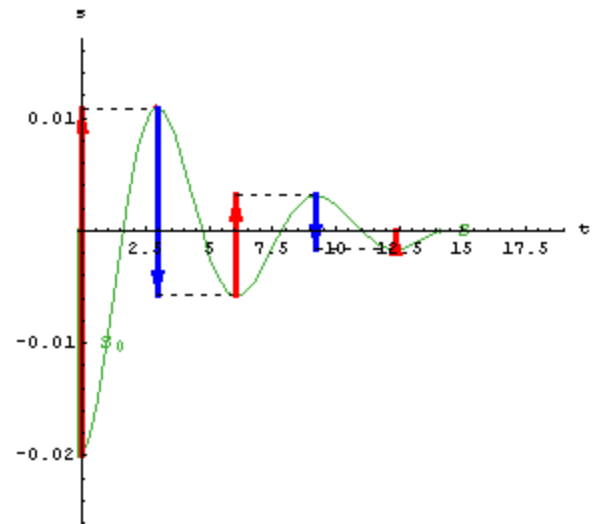
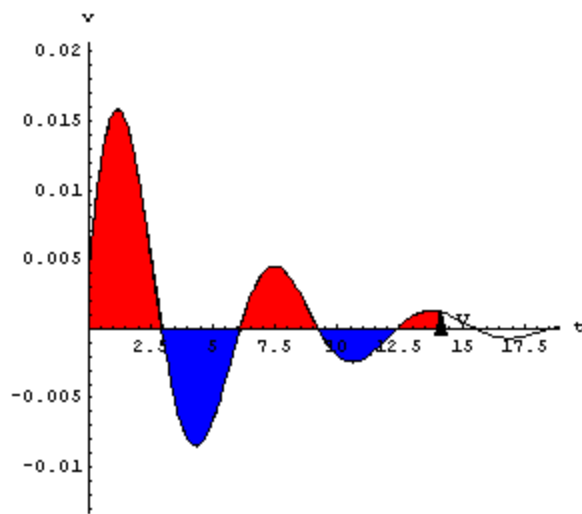


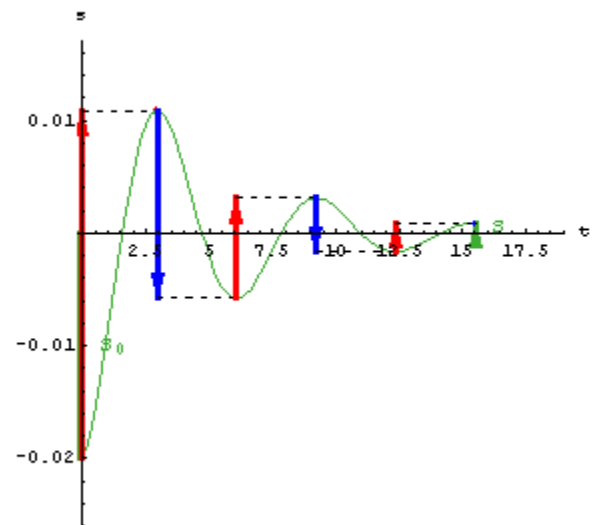
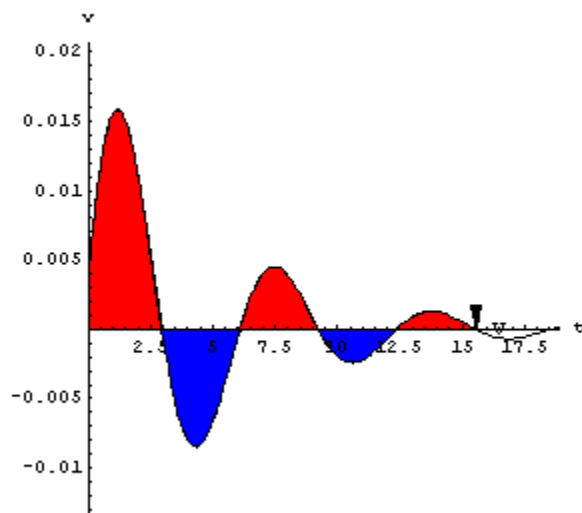
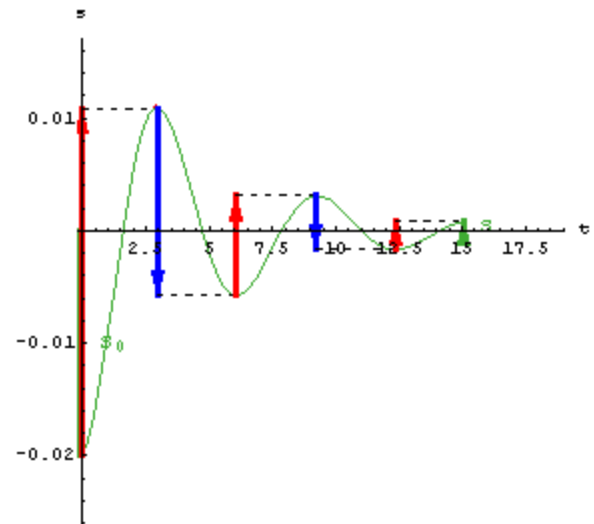
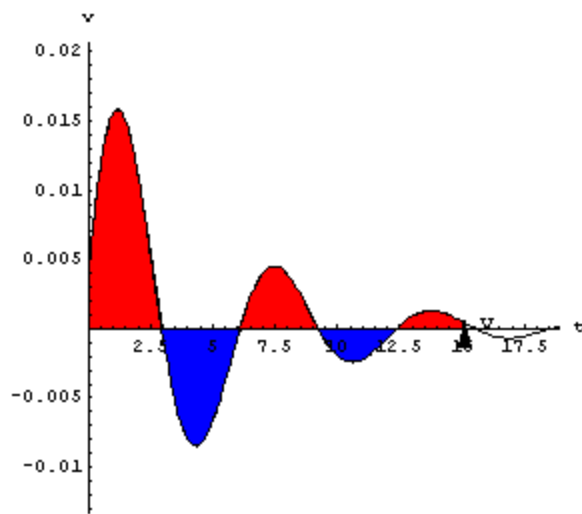


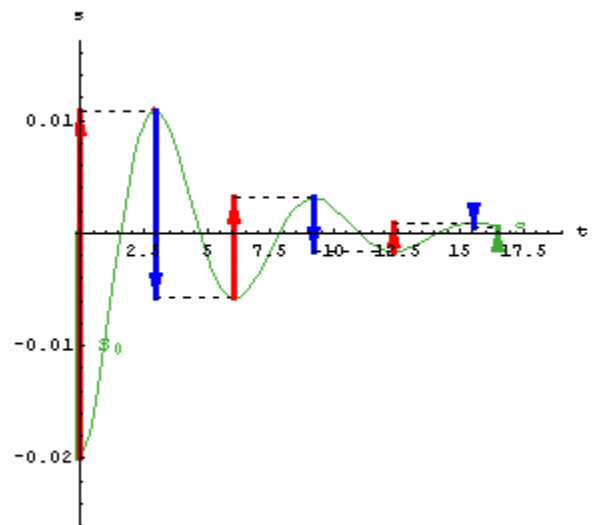
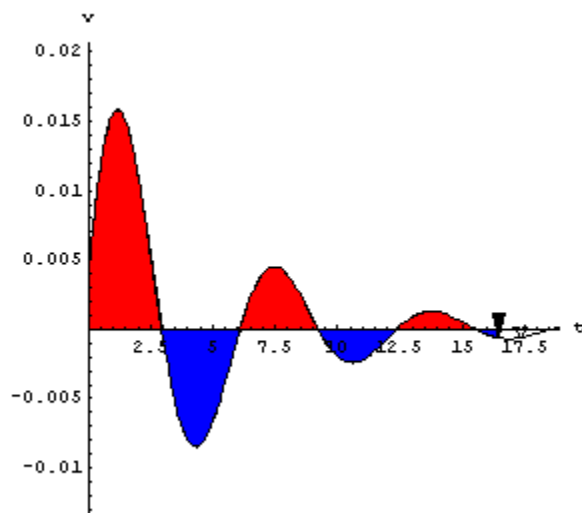
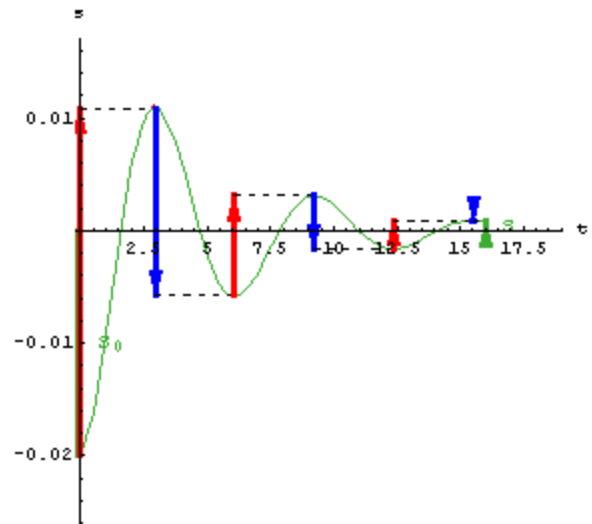
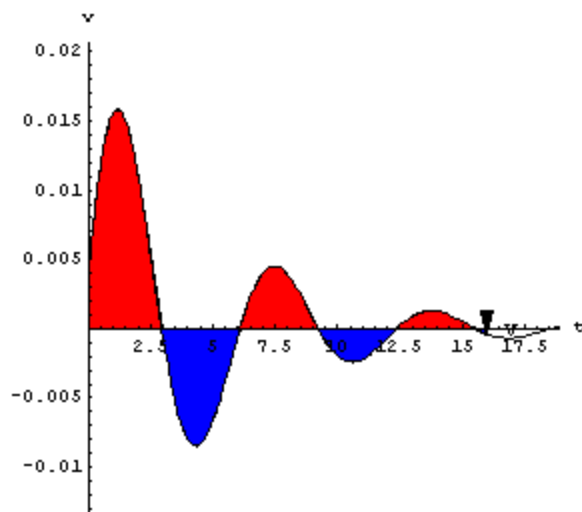


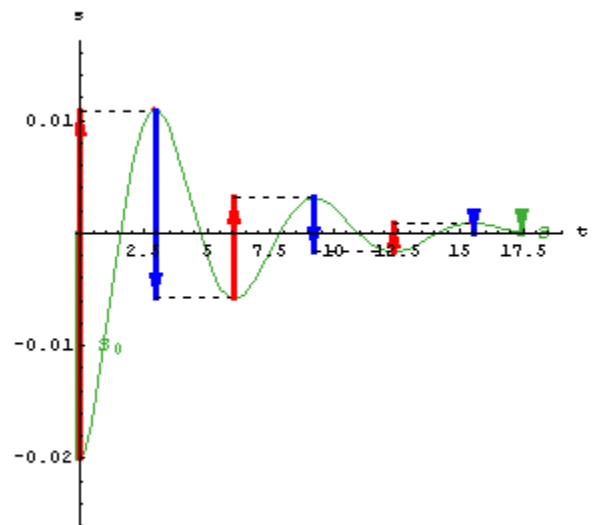
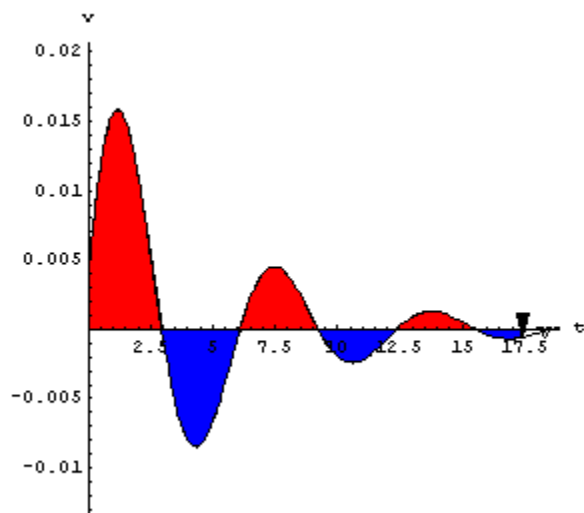
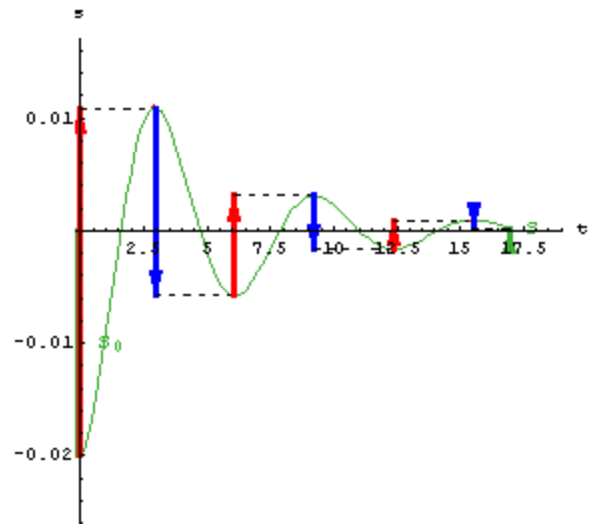
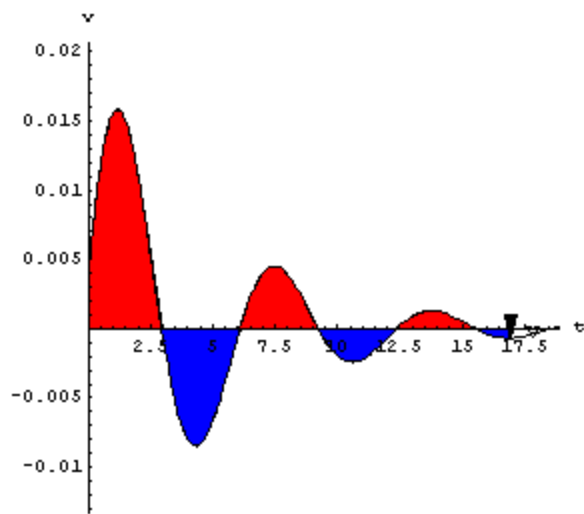


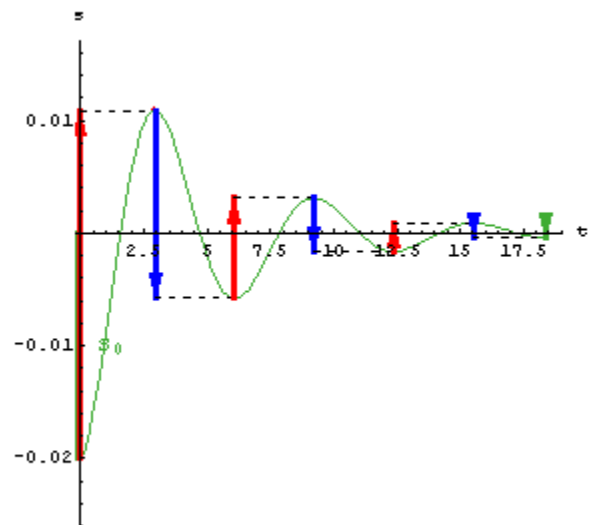
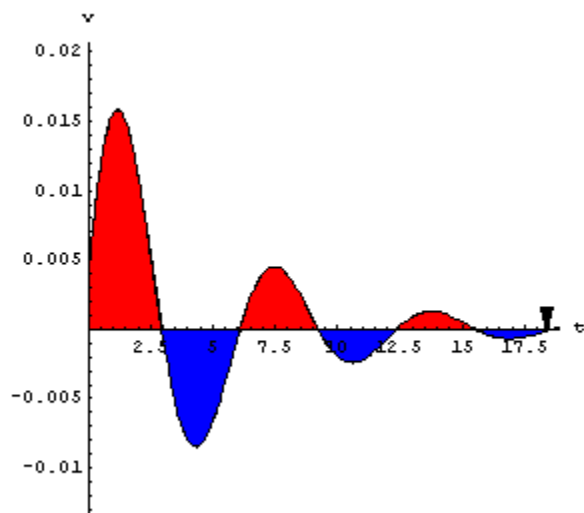
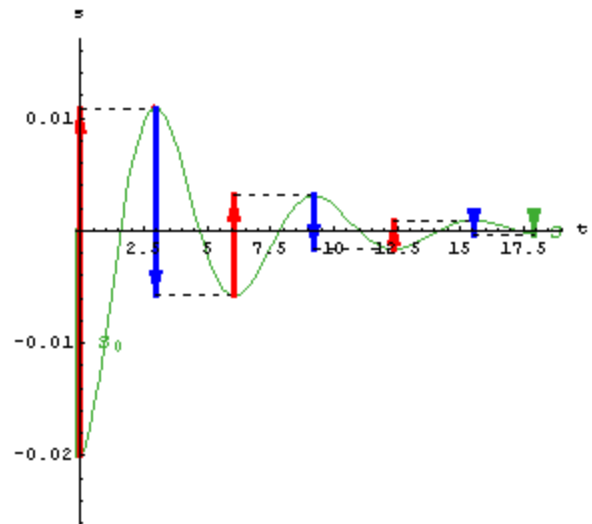
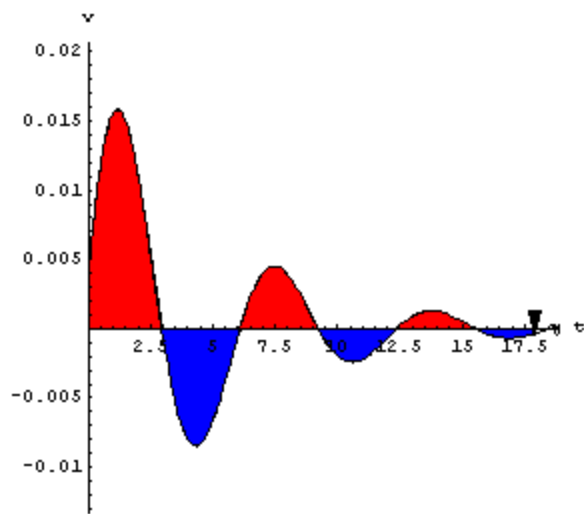


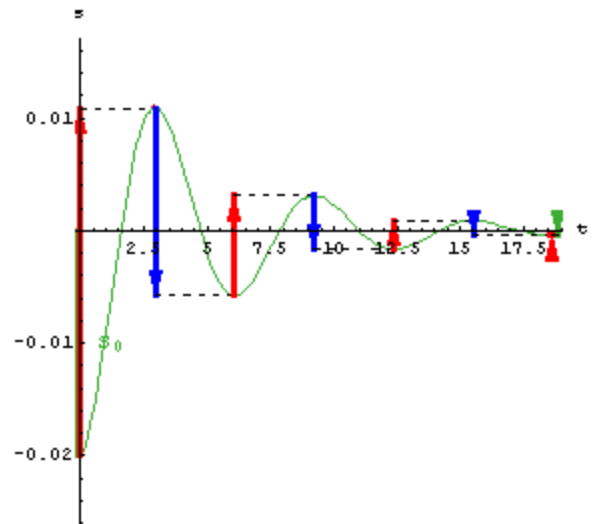
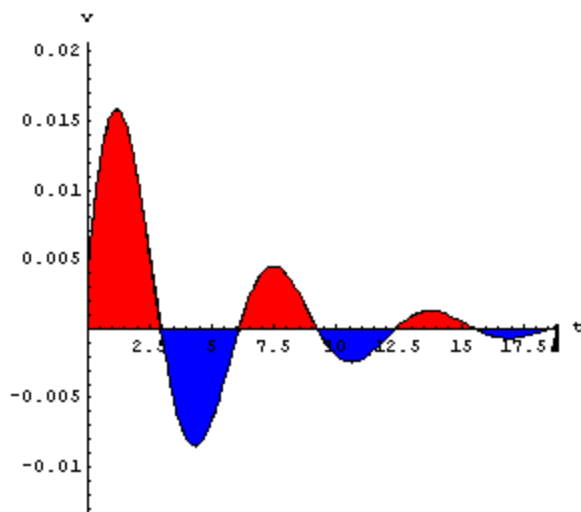






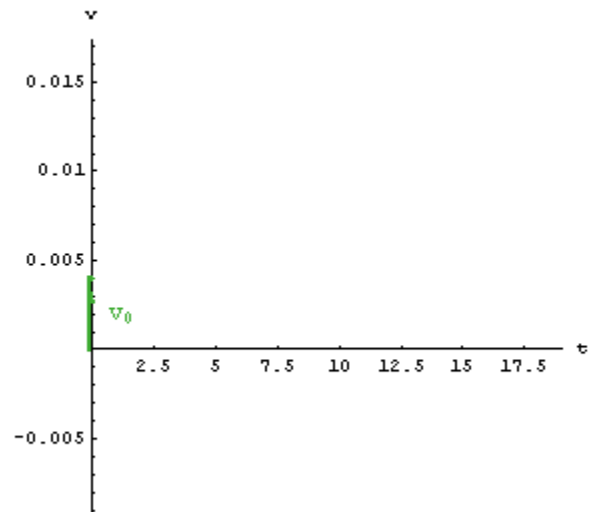
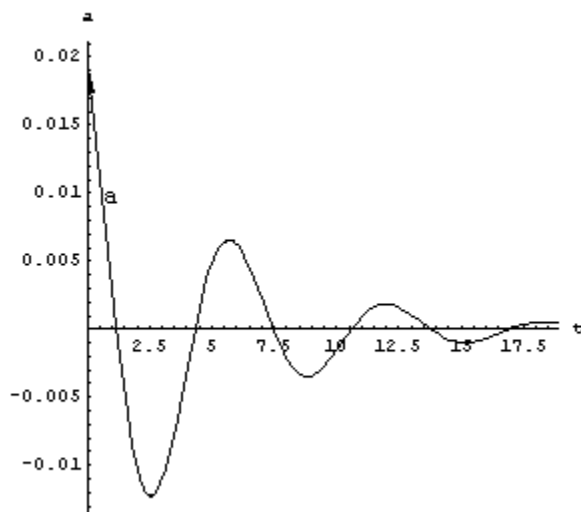


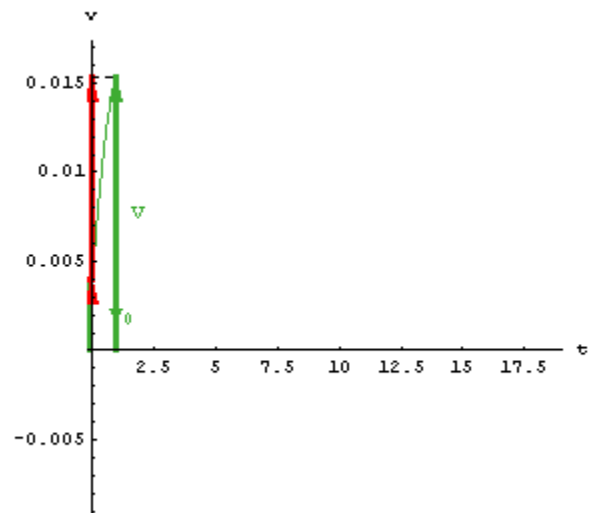
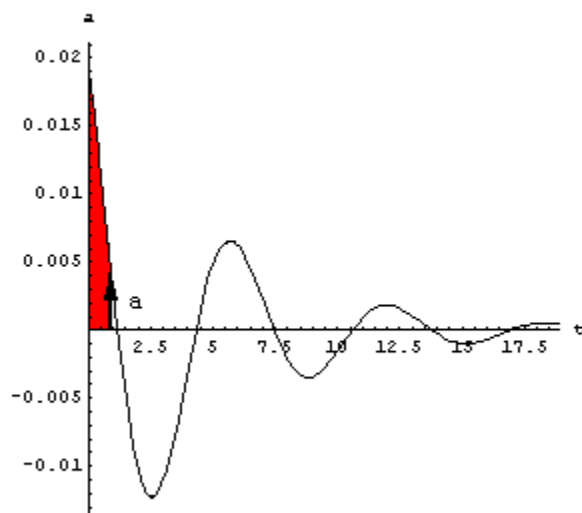
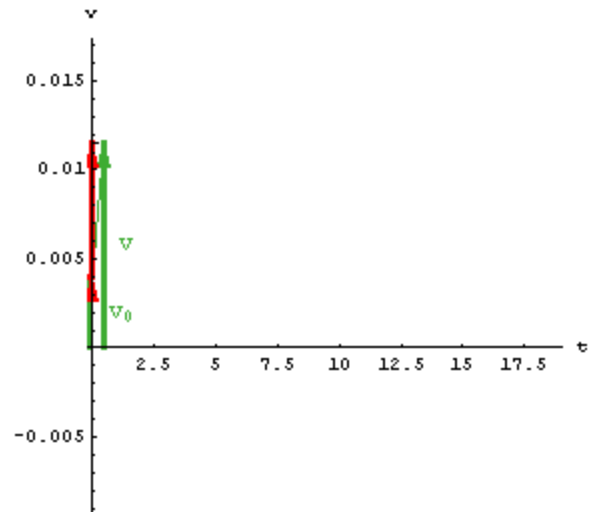
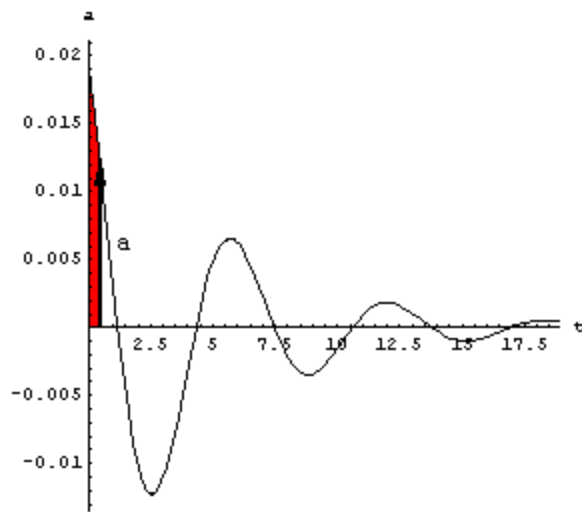


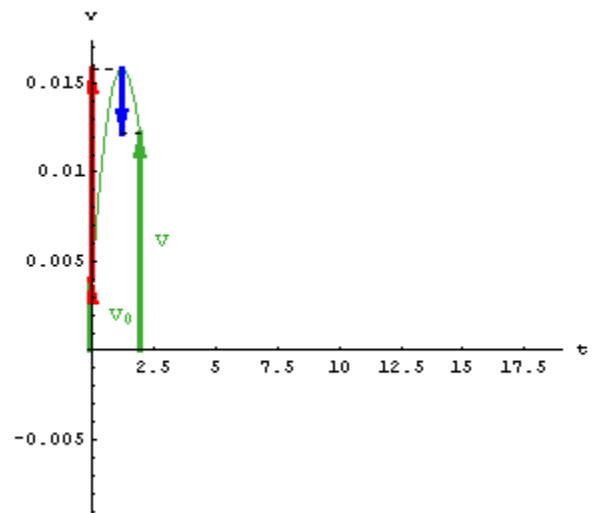
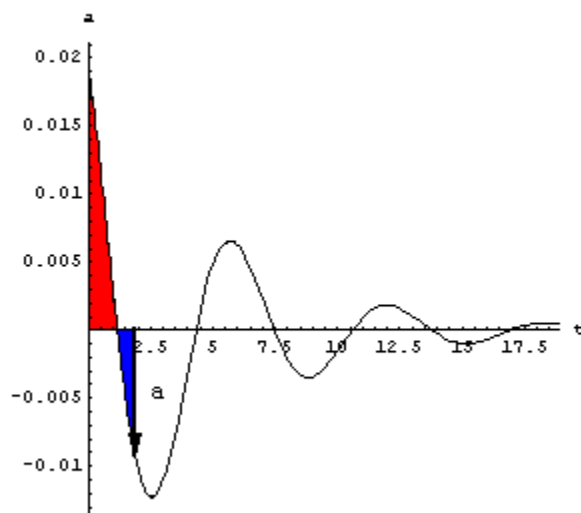
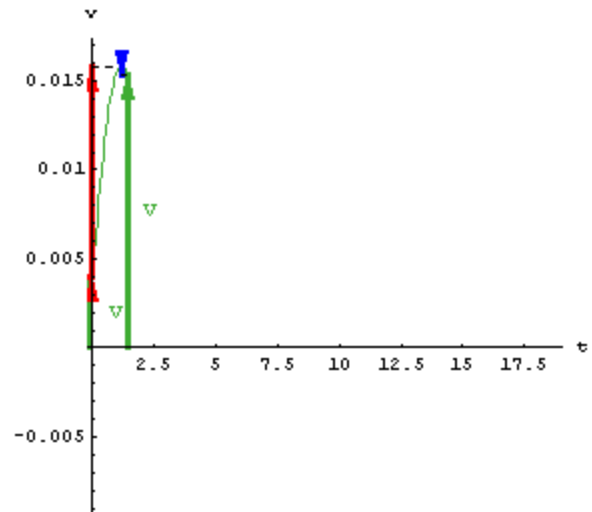
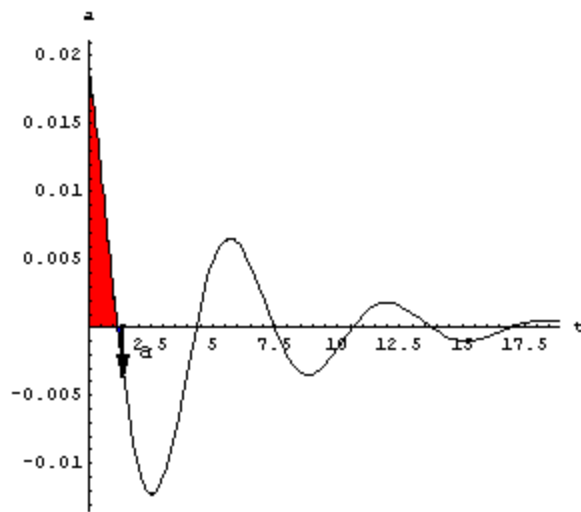


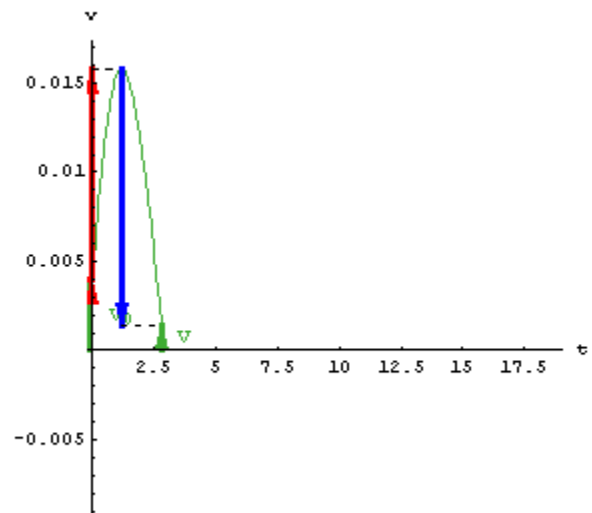
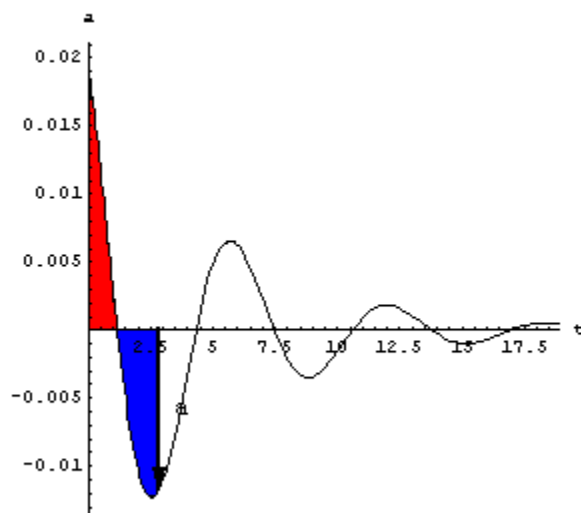
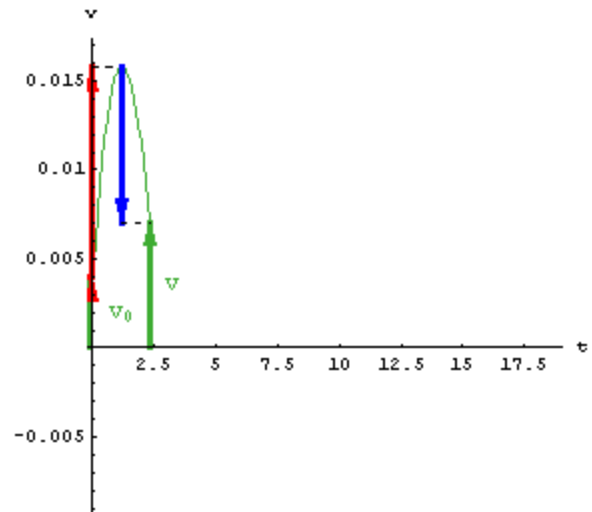
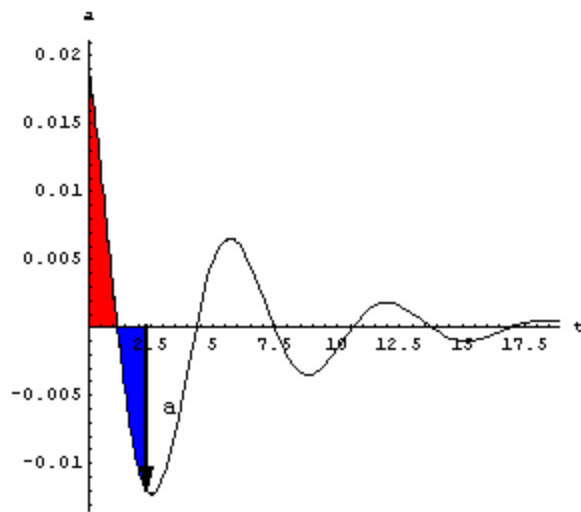
In[54]:=

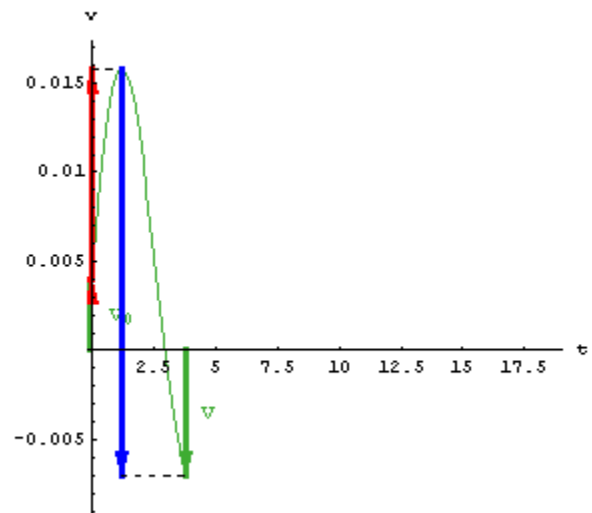
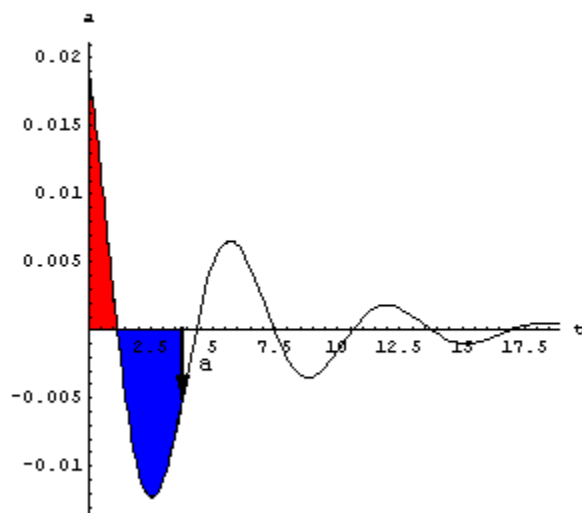
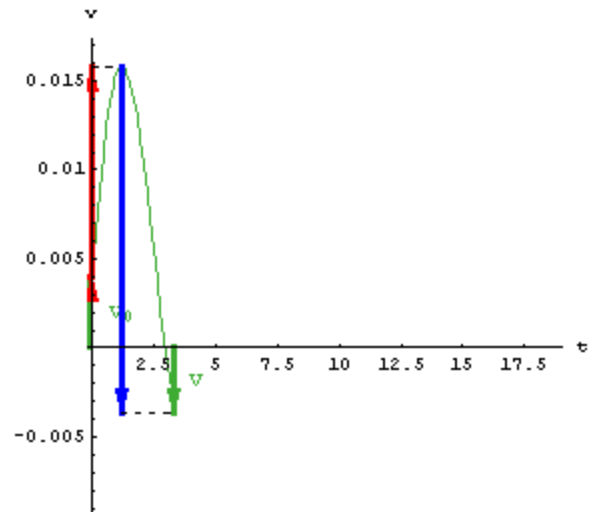
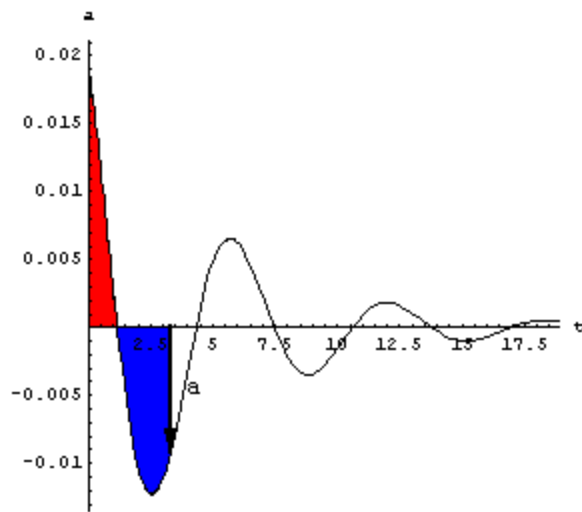
```
velocity[a[t], {t, 0, 6 * Pi}, v[0]];
```

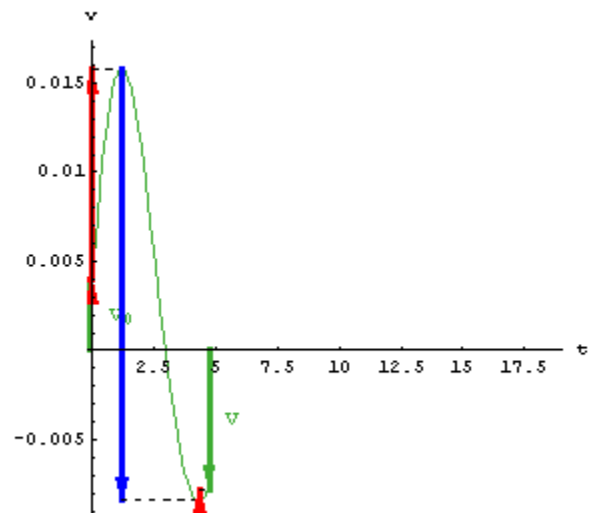
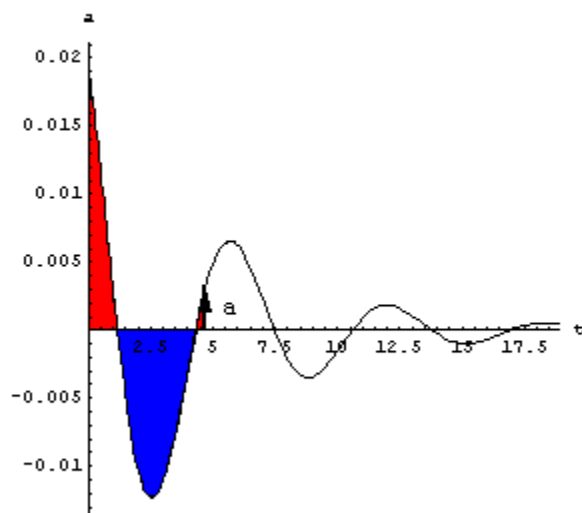
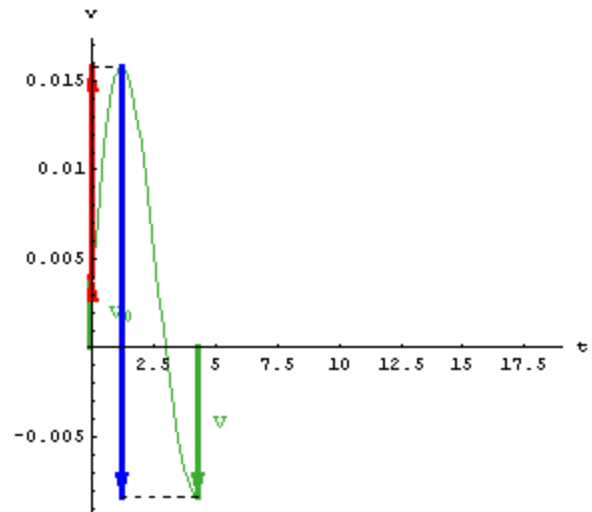
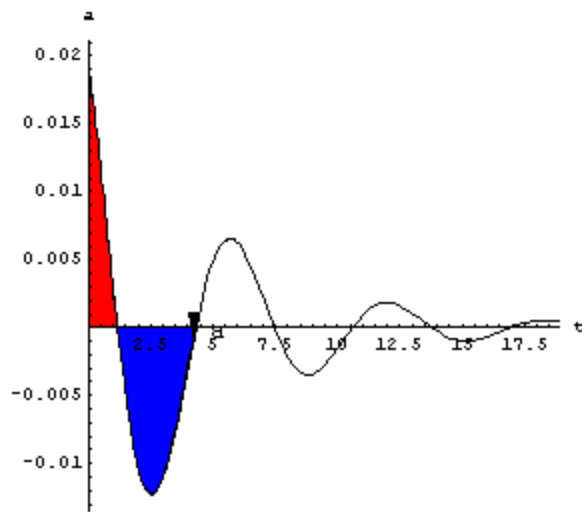


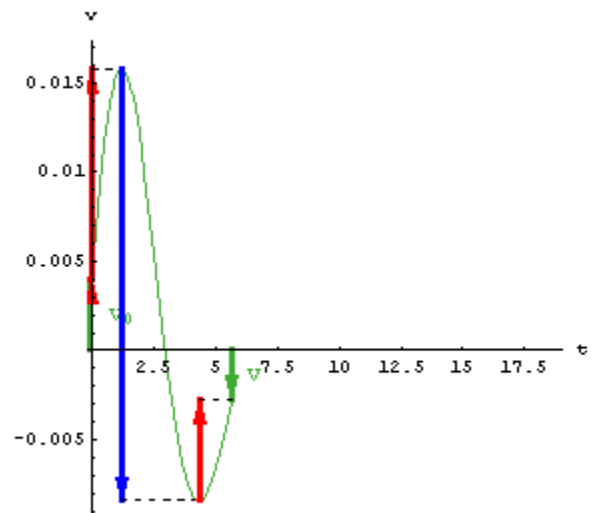
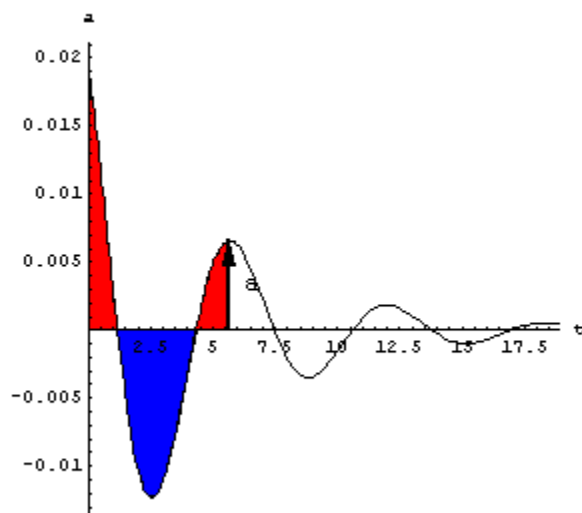
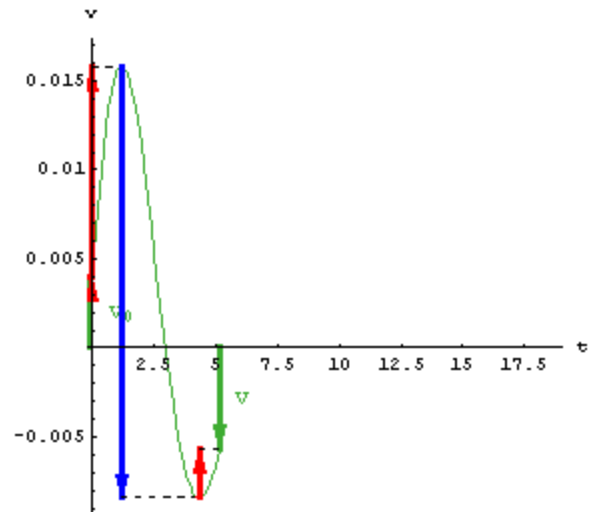
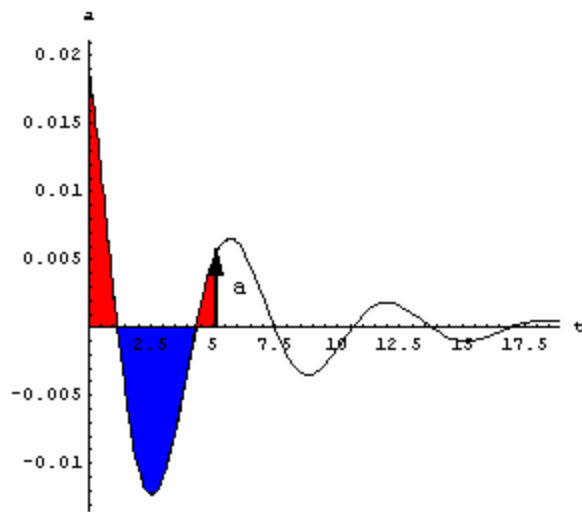


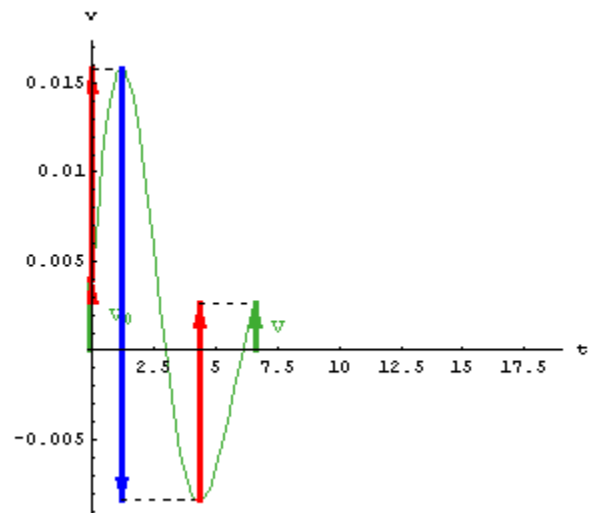
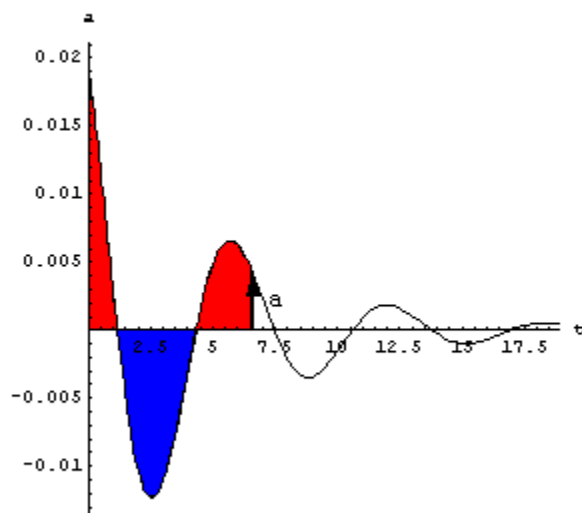
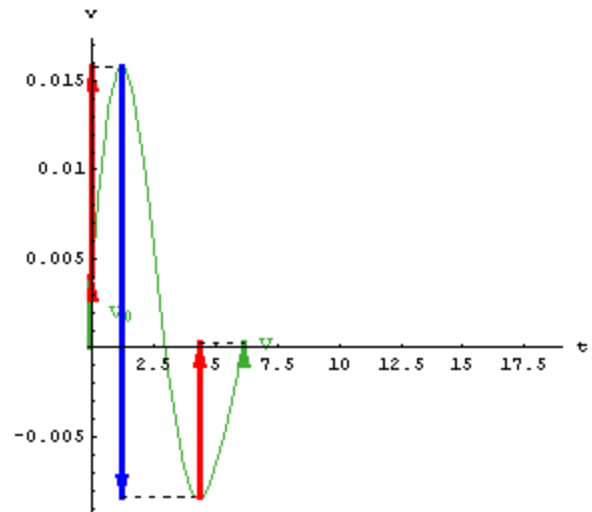
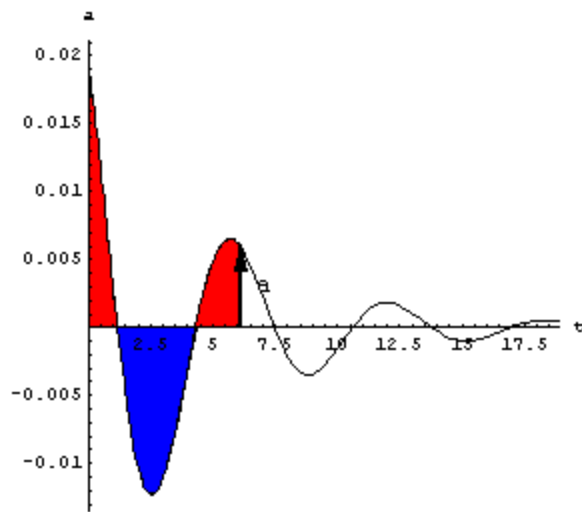


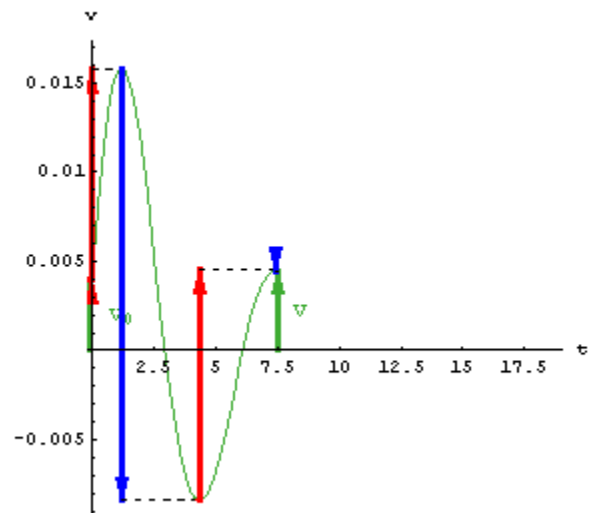
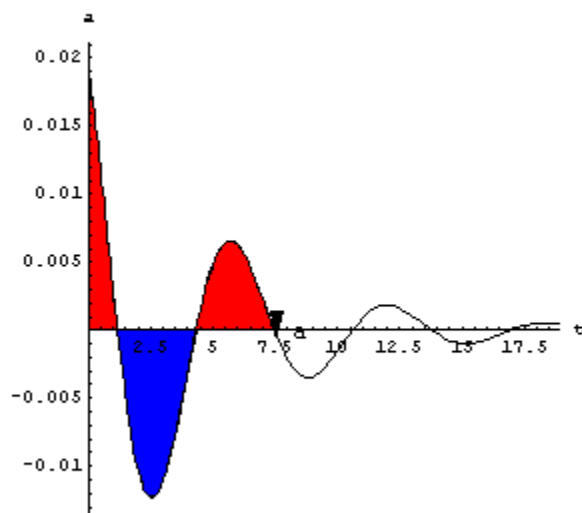
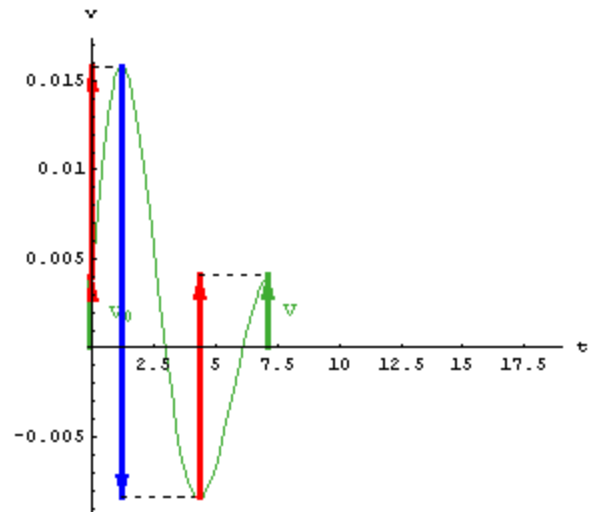
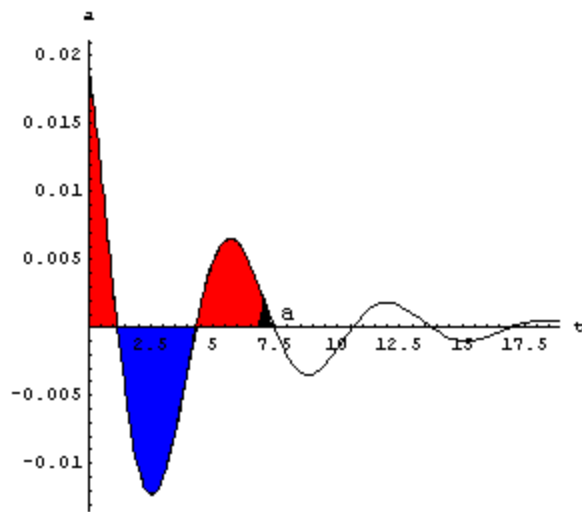


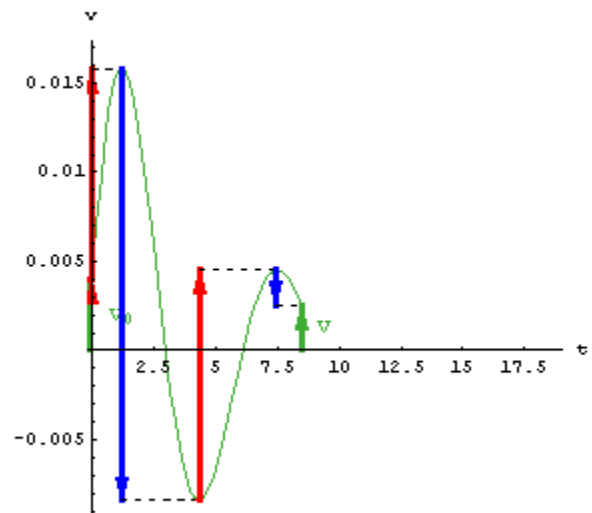
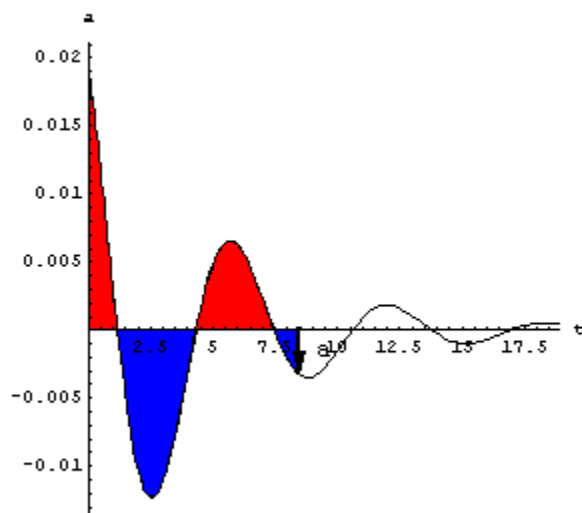
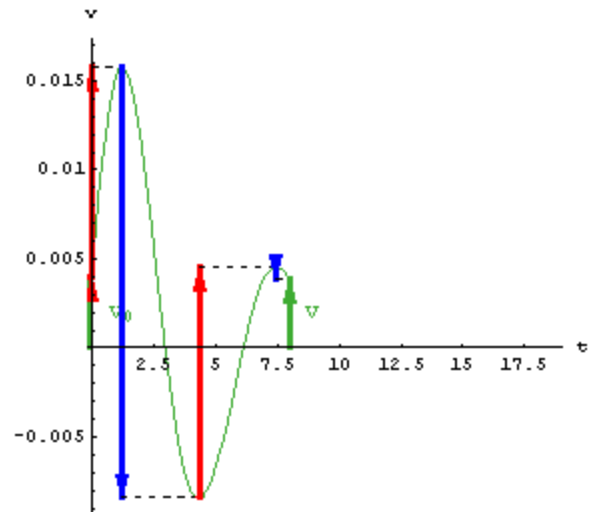
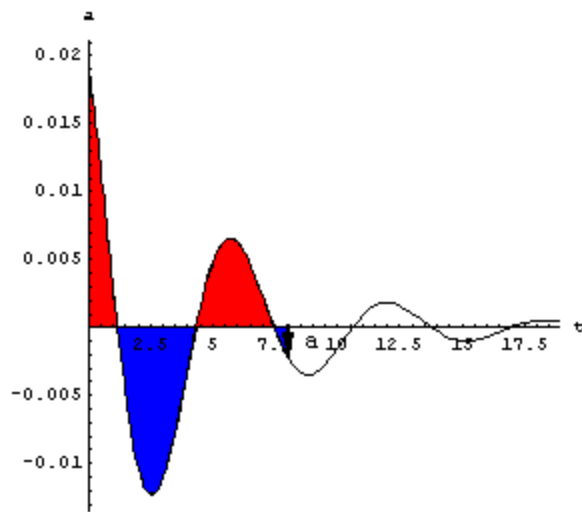


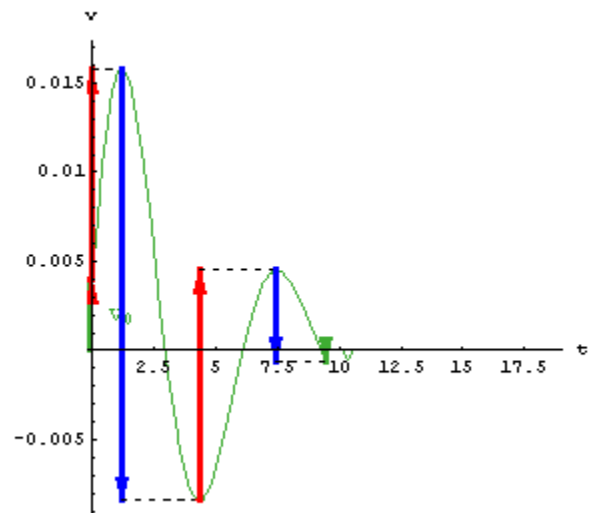
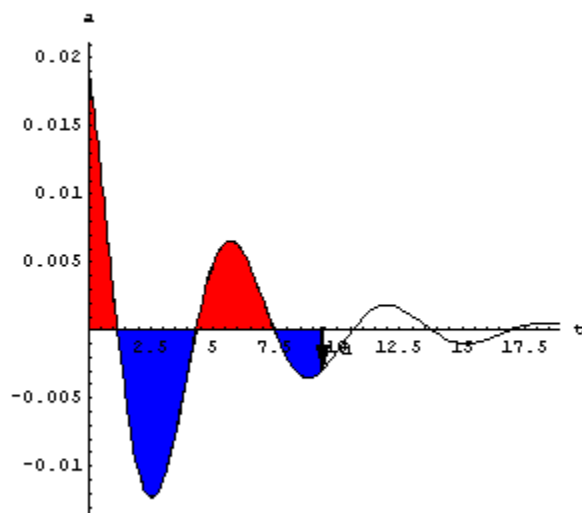
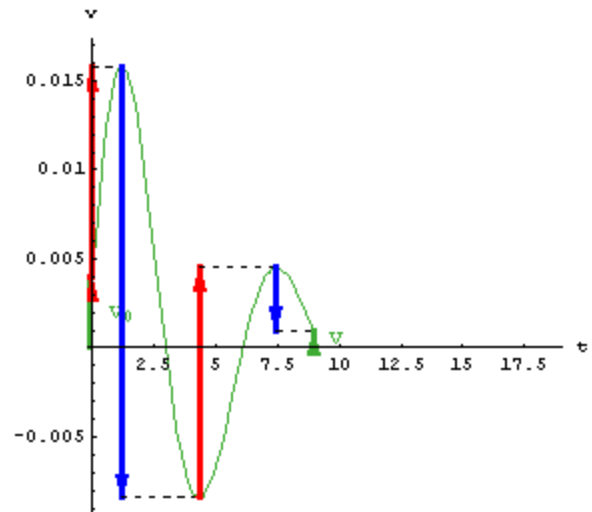
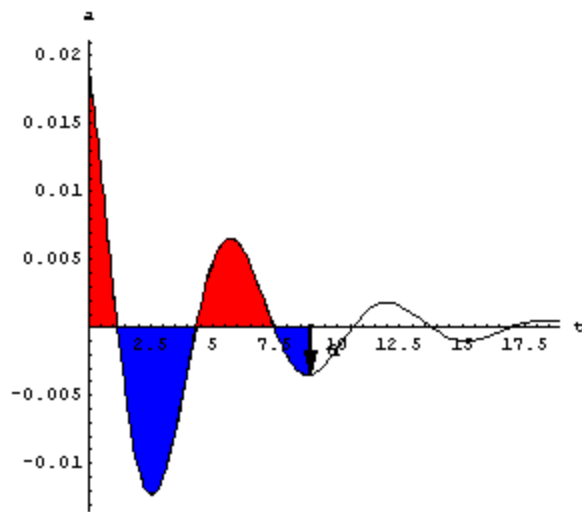


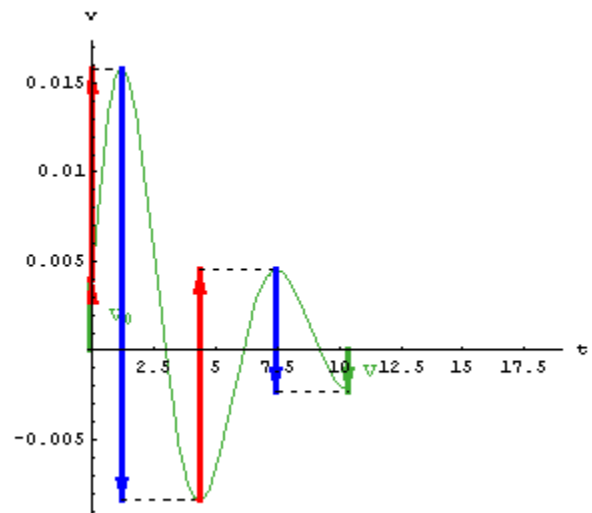
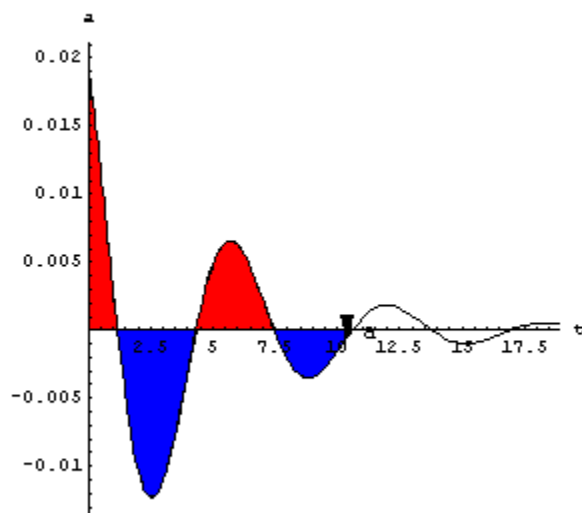
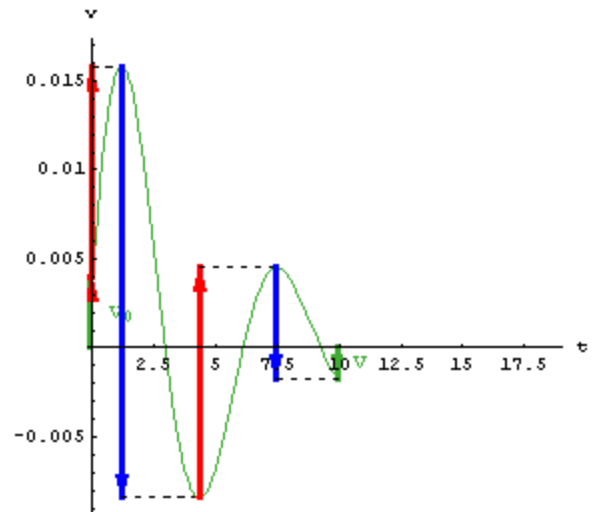
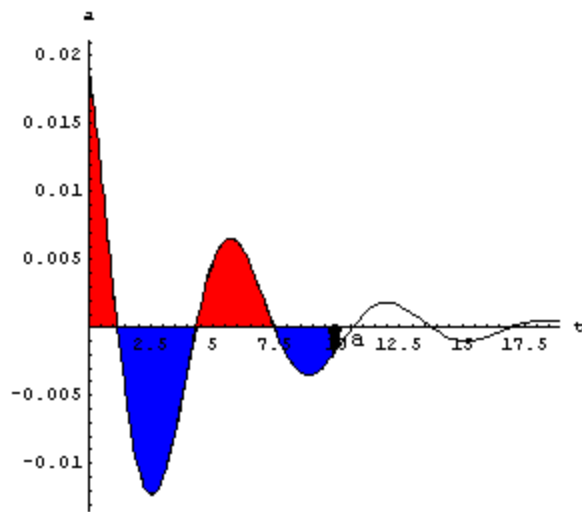


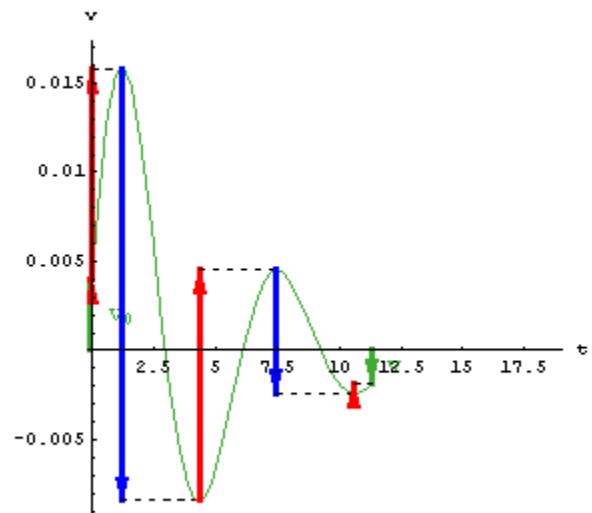
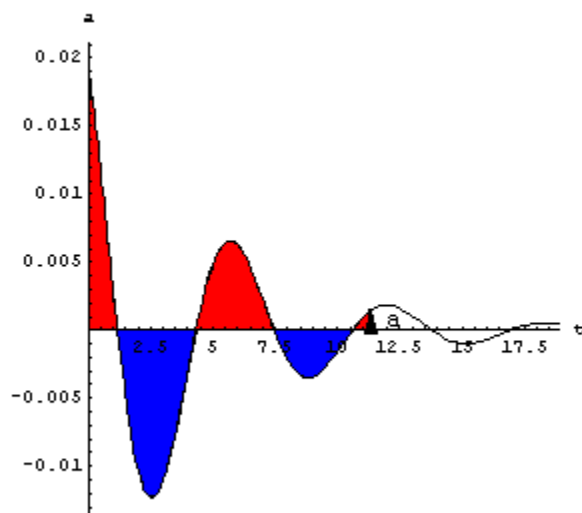
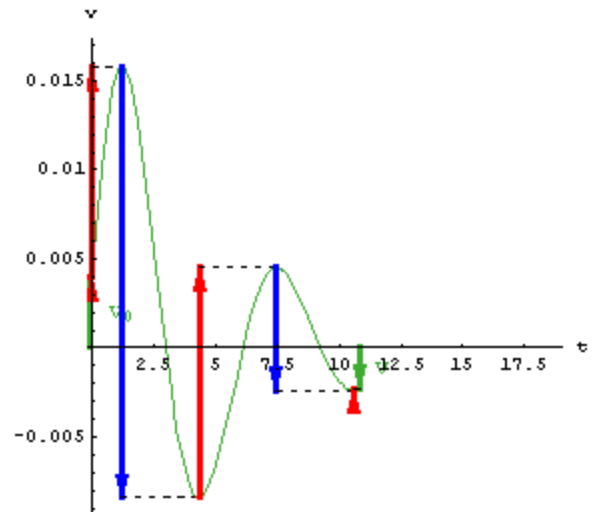
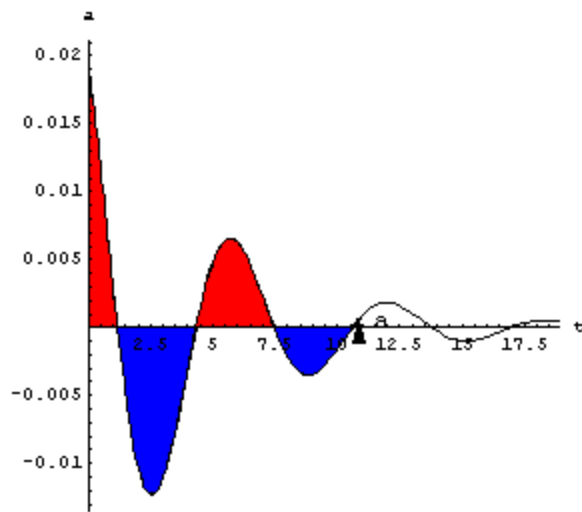


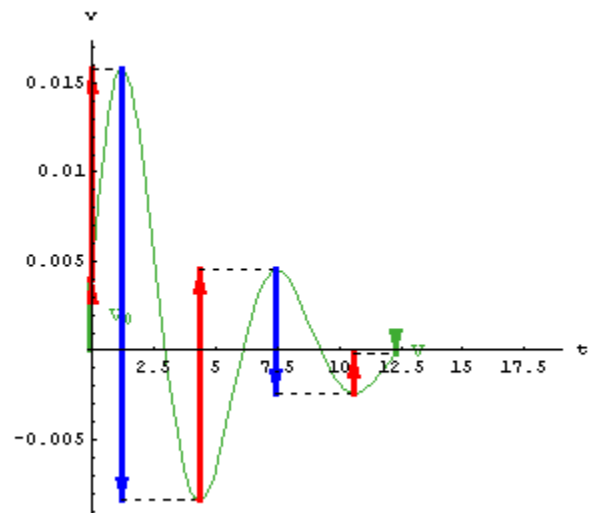
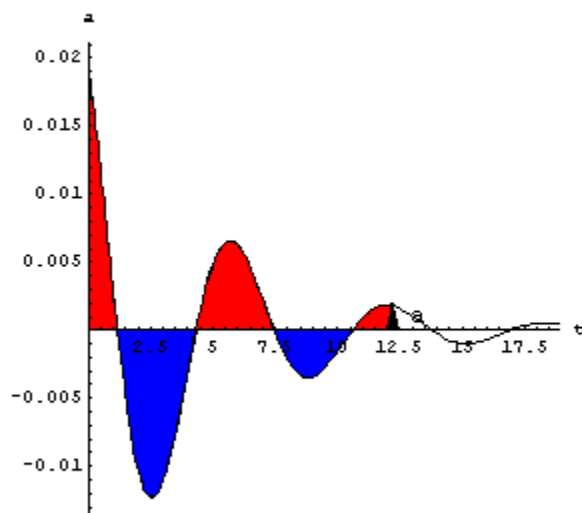
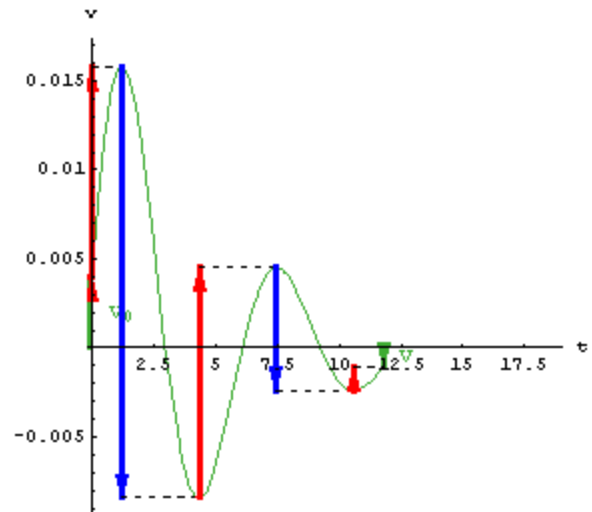
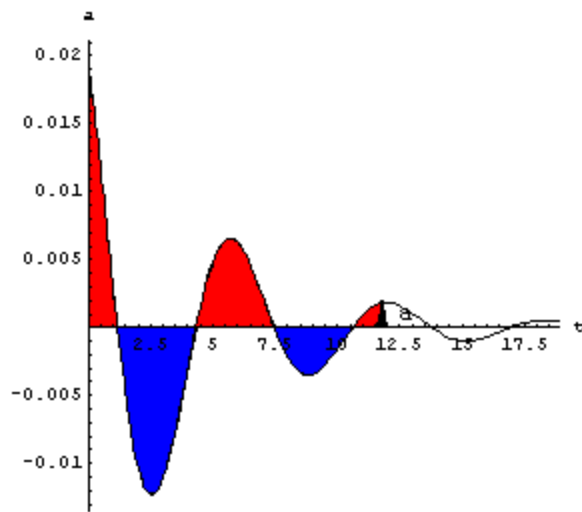


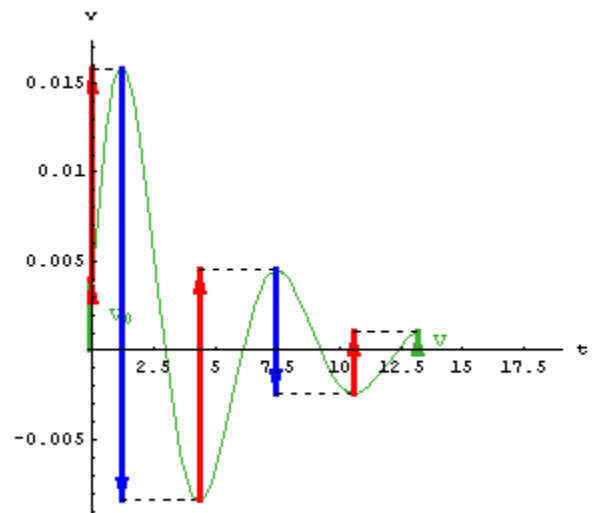
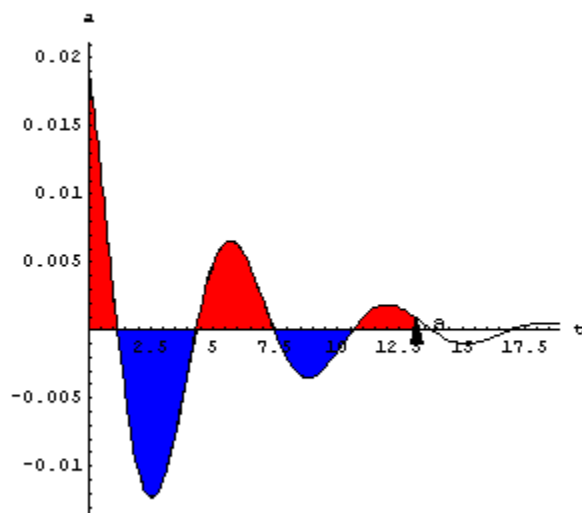
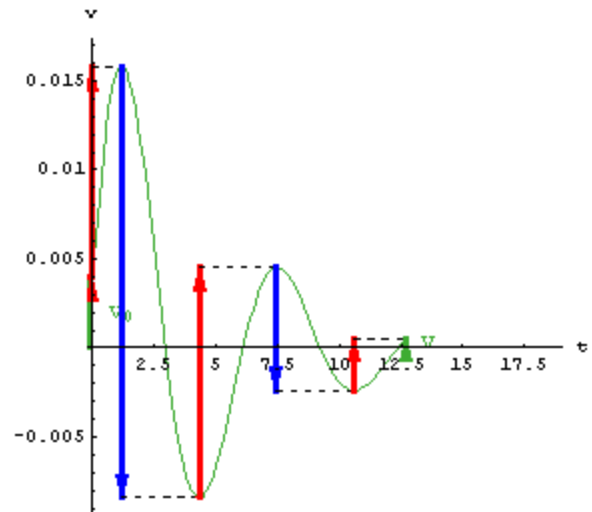
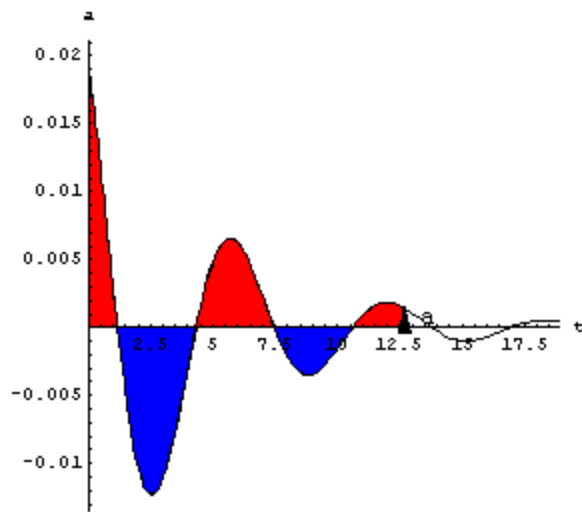


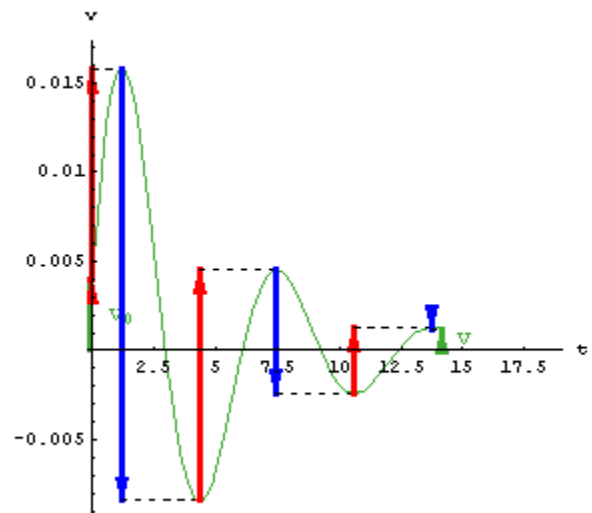
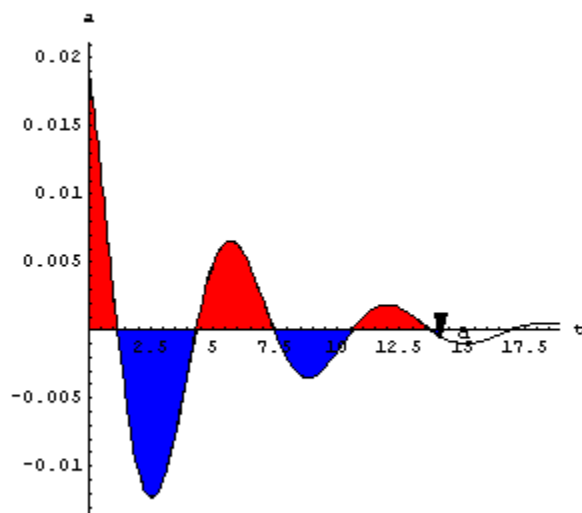
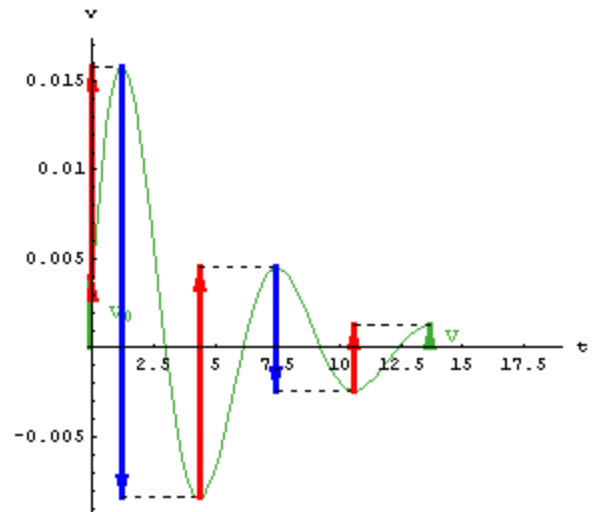
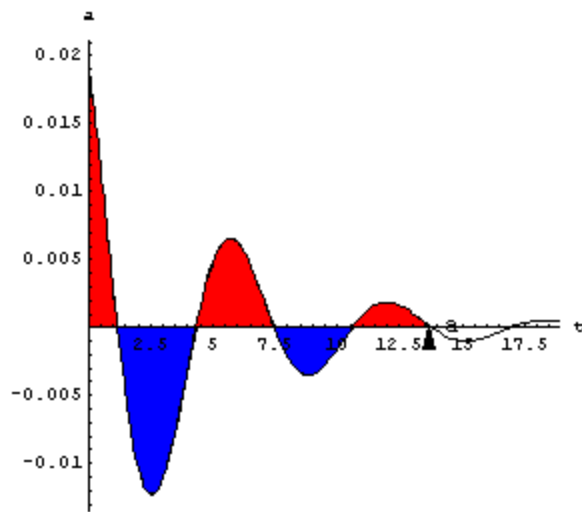


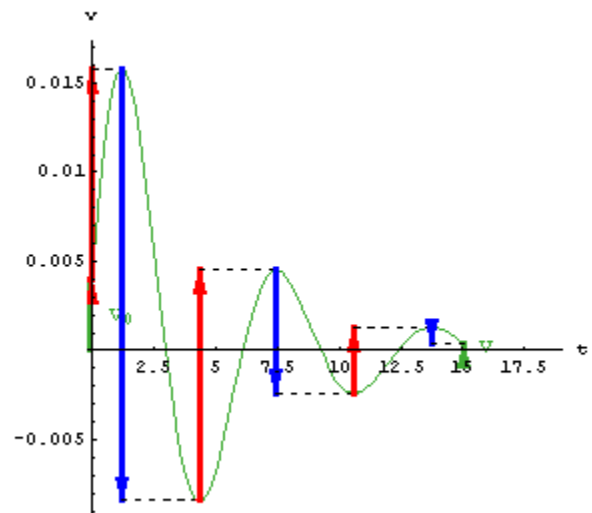
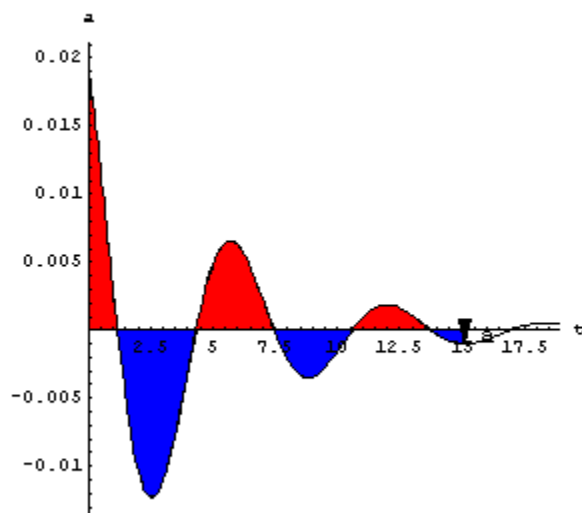
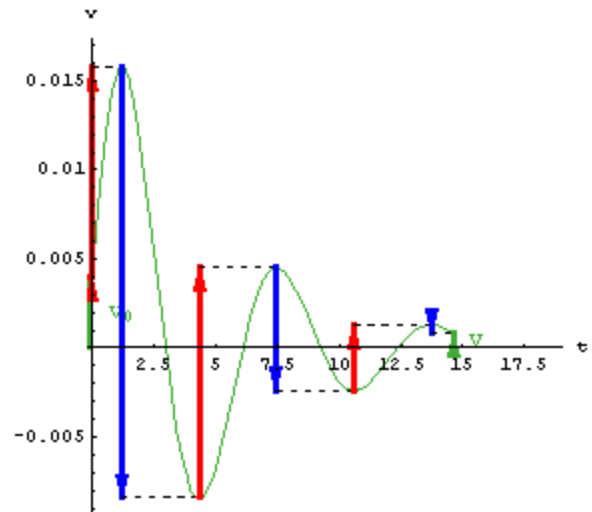
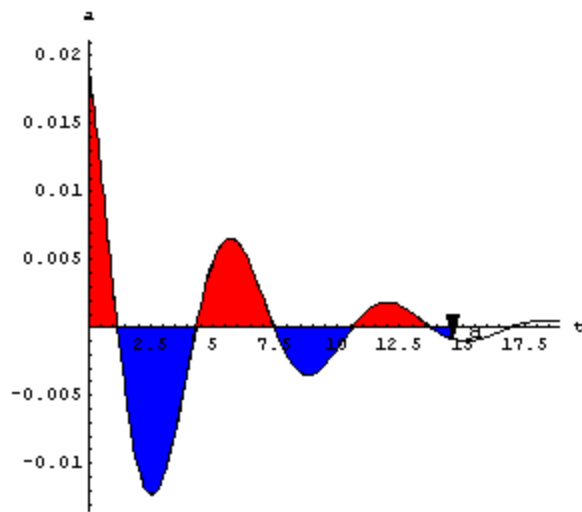


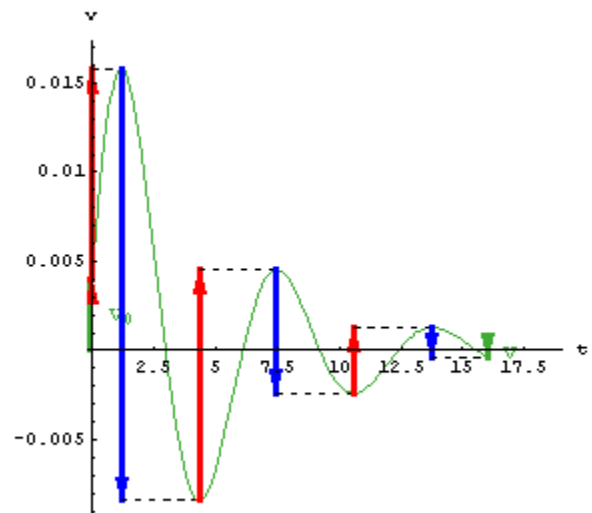
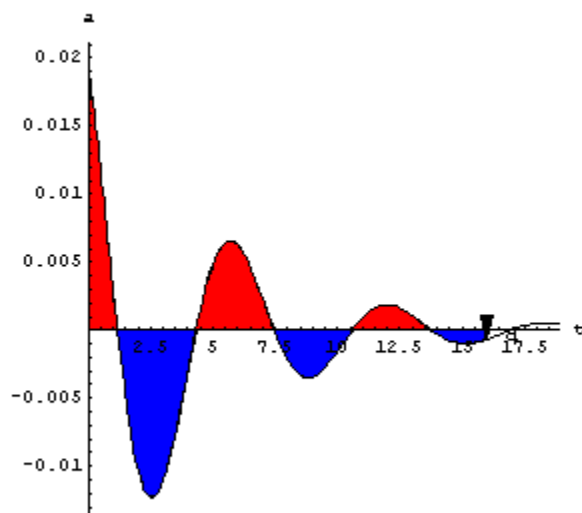
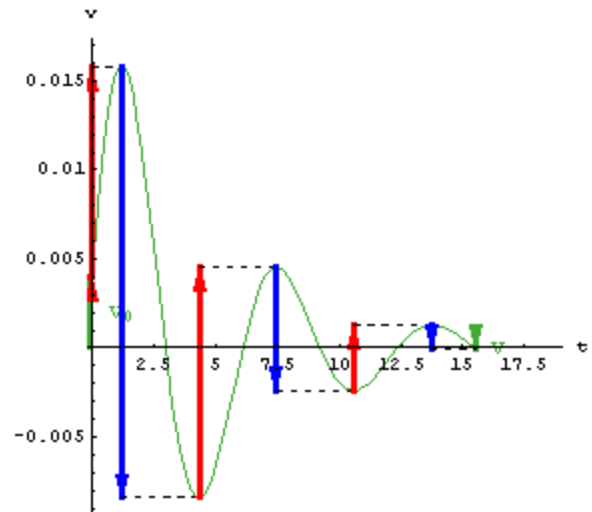
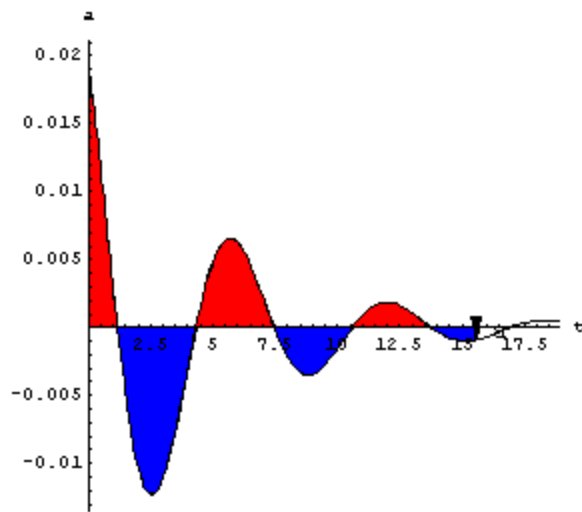


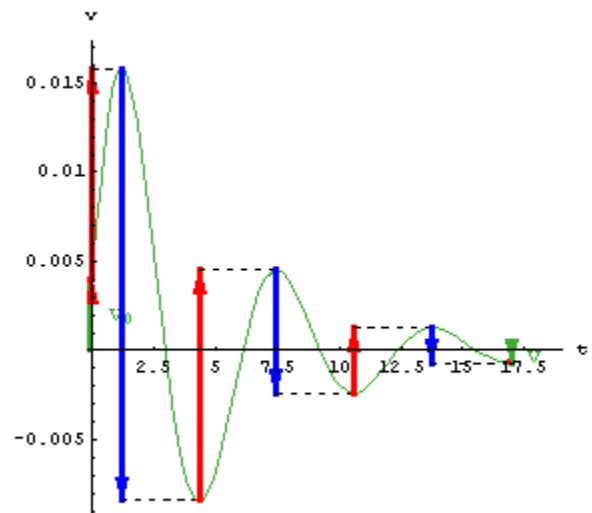
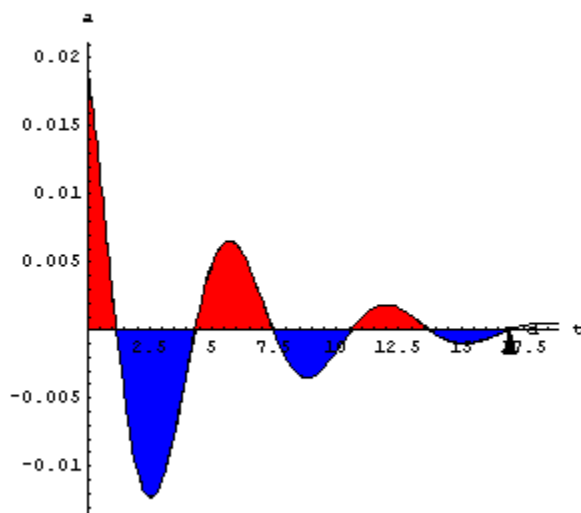
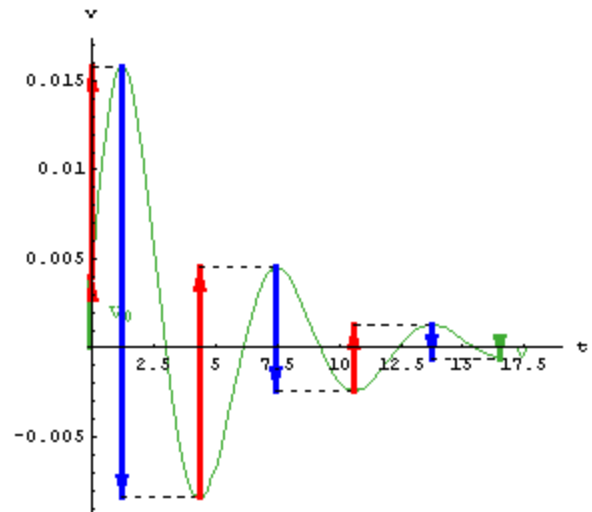
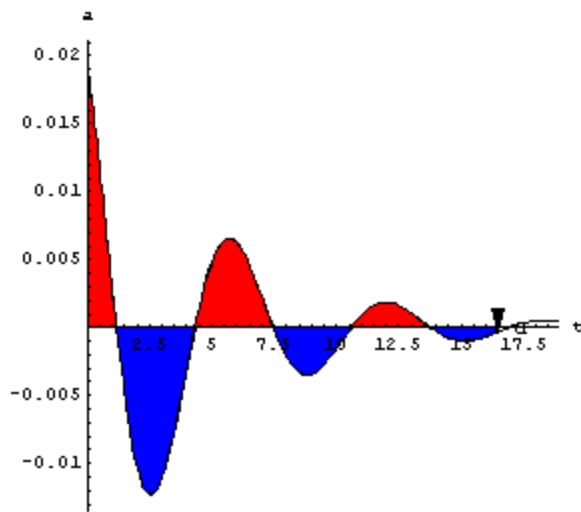


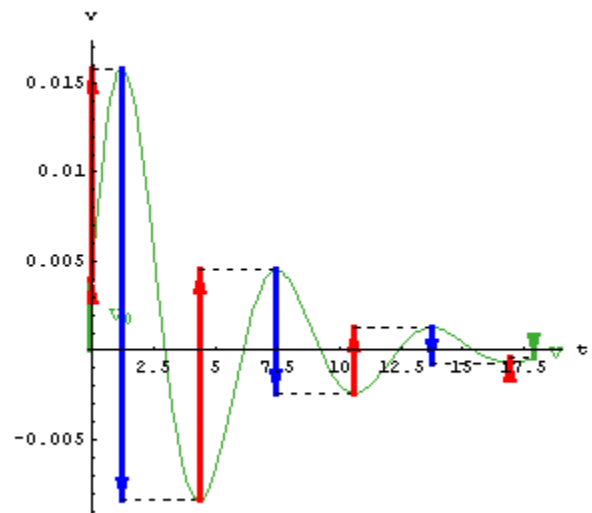
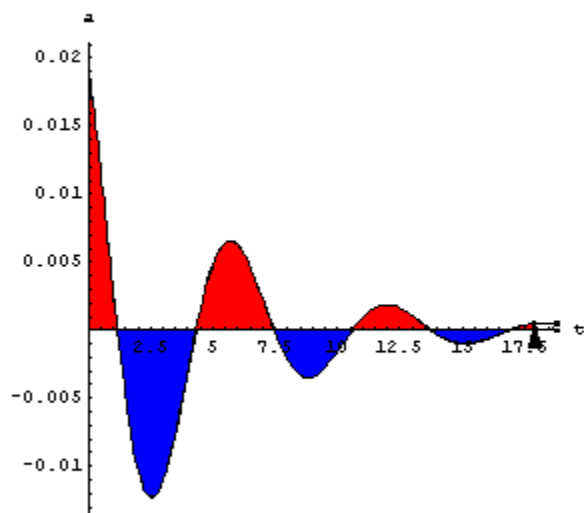
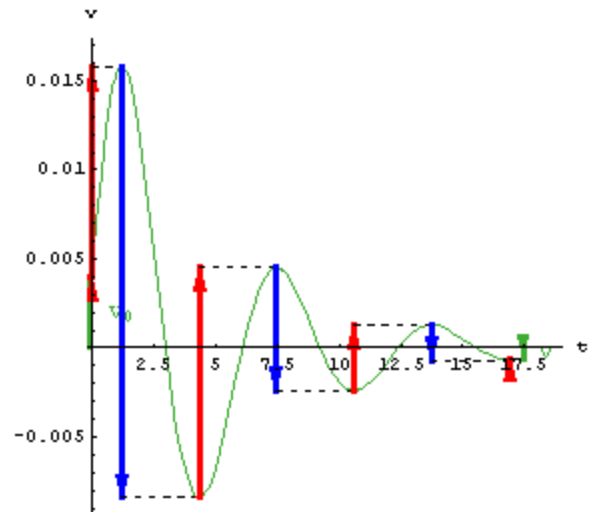
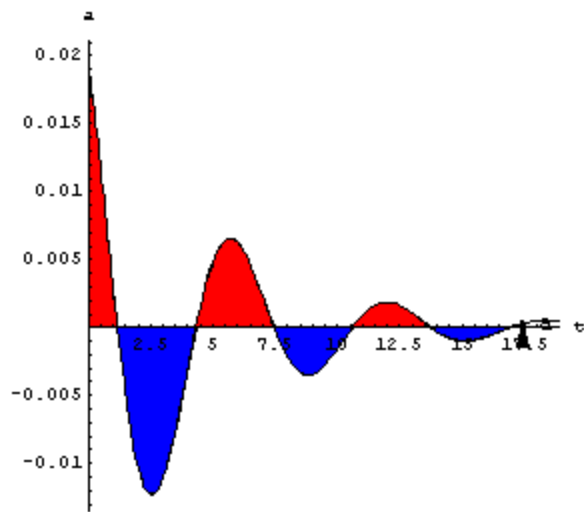


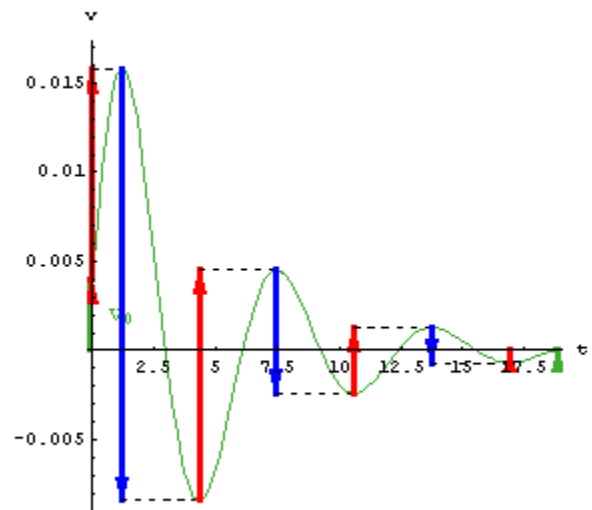
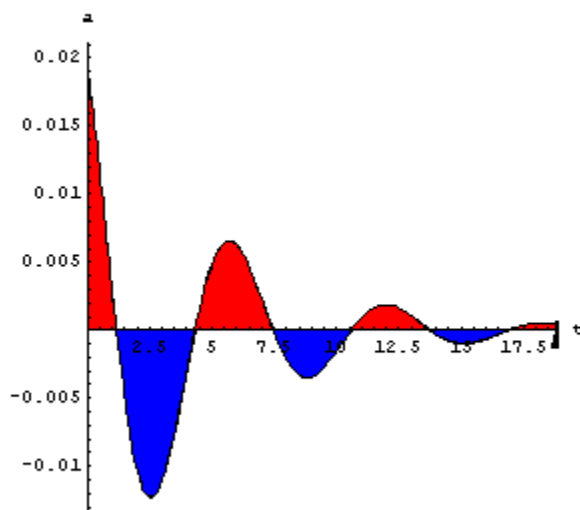
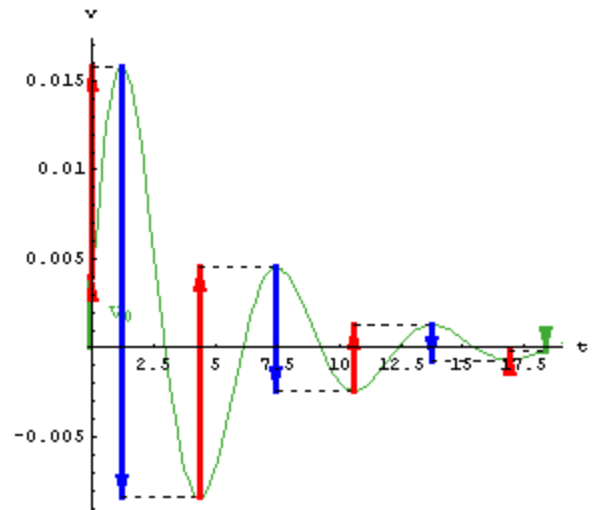
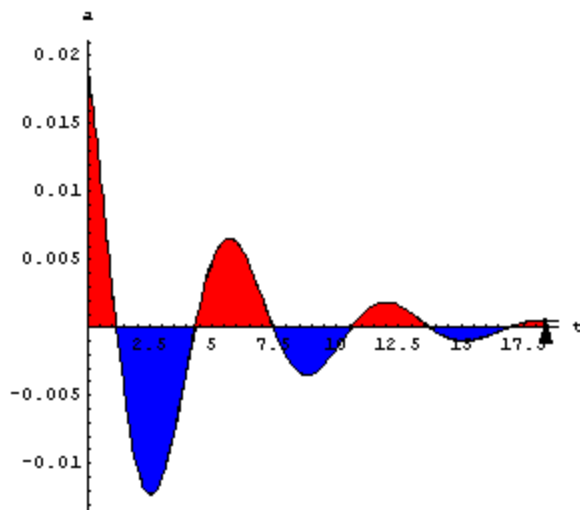












You Try It: A Kid on a Swing

While friction takes energy out of an oscillating mass and spring, there are some systems where energy is continually added to the mass so that the oscillations actually get bigger and bigger in amplitude. An example of this type of system is a kid on a swing, "pumping" to get the swing going or being pushed by another kid. In this situation, the position starts at zero and the amplitude of the oscillation gets bigger as time progresses.

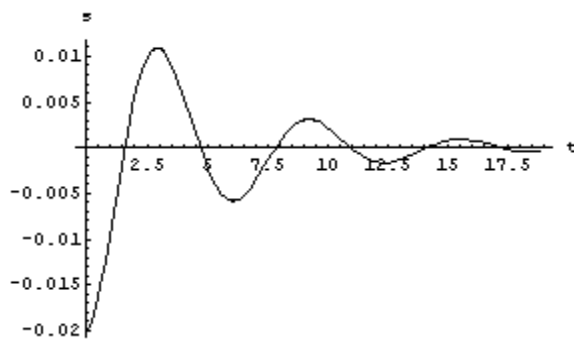
1. Modify the code below to make a model of this kind of motion. (Hint: The $\sin t$ is 0 at $t=0$, and the oscillations should grow rather than decay.)

In[55]:=

```
Clear[s];

s[t_] = -0.02 * Exp[-t / 5] * Cos[t];

Plot[s[t], {t, 0, 6 * Pi}, PlotRange -> All,
  AxesLabel -> {"t", "s"}];
```



2. Find the velocity and acceleration of the kid, and plot them.

In[58]:=

```
v[t_] = D[s[t], t]
```

Out[58]=

```
0.004 e-t/5 Cos[t] + 0.02 e-t/5 Sin[t]
```

In[59]:=

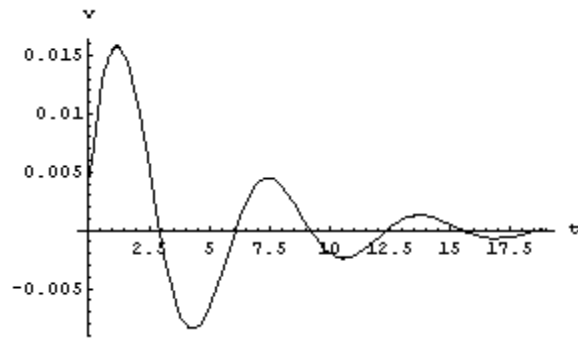
```
a[t_] = D[v[t], t]
```

Out[59]=

```
0.0192 e-t/5 Cos[t] - 0.008 e-t/5 Sin[t]
```

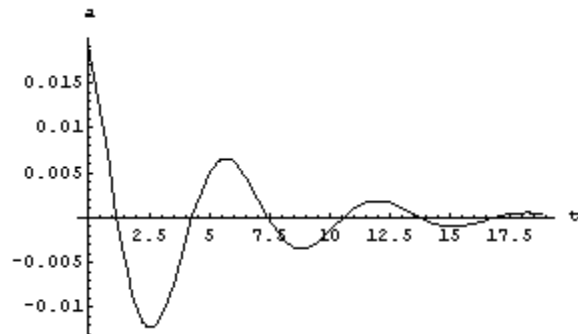
In[60]:=

```
Plot[v[t], {t, 0, 6 * Pi}, PlotRange -> All,
  AxesLabel -> {"t", "v"}];
```



In[61]:=

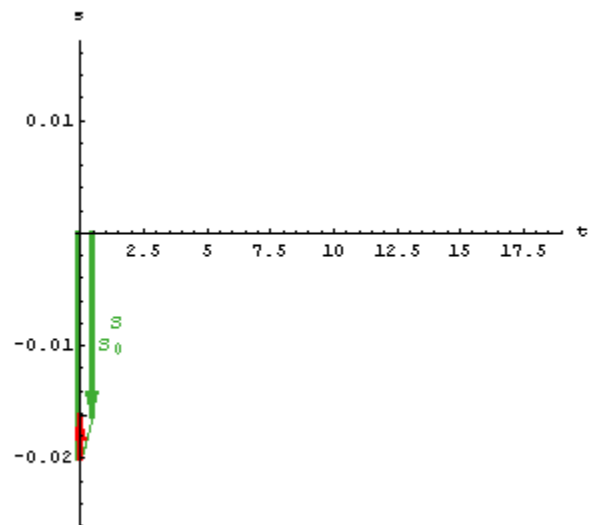
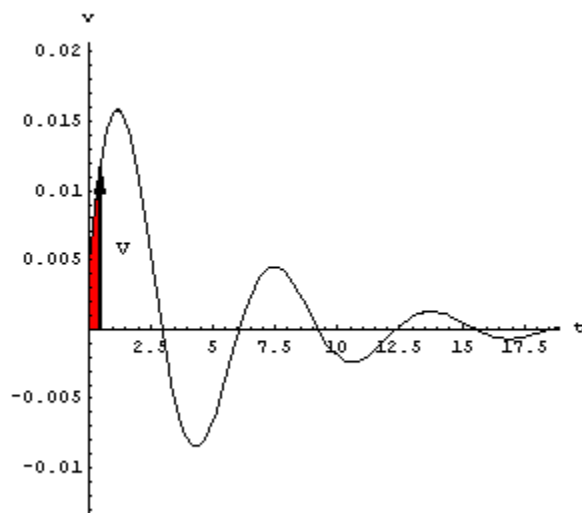
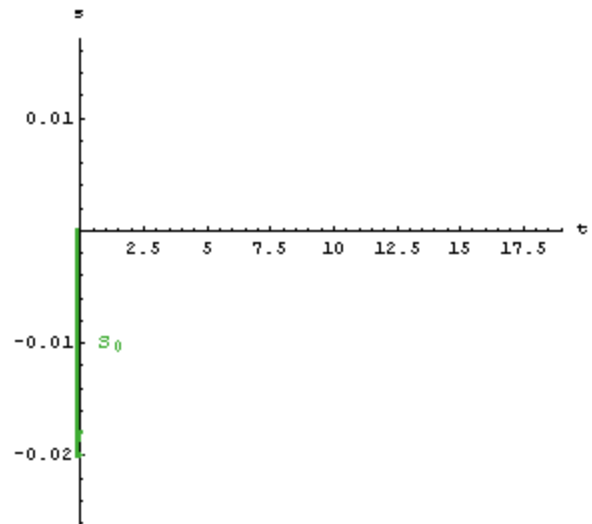
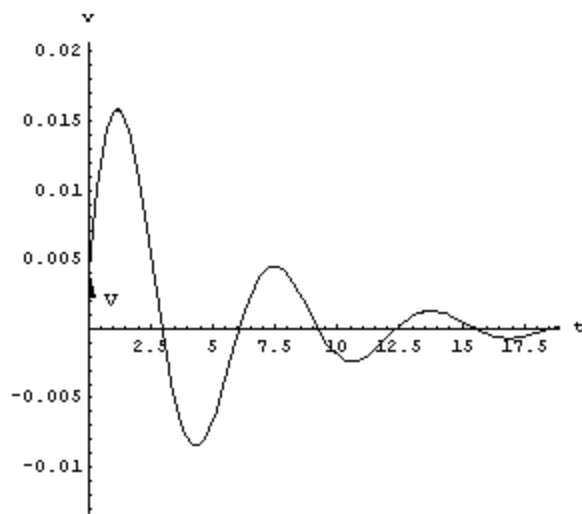
```
Plot[a[t], {t, 0, 6 * Pi}, PlotRange -> All,
  AxesLabel -> {"t", "a"}];
```

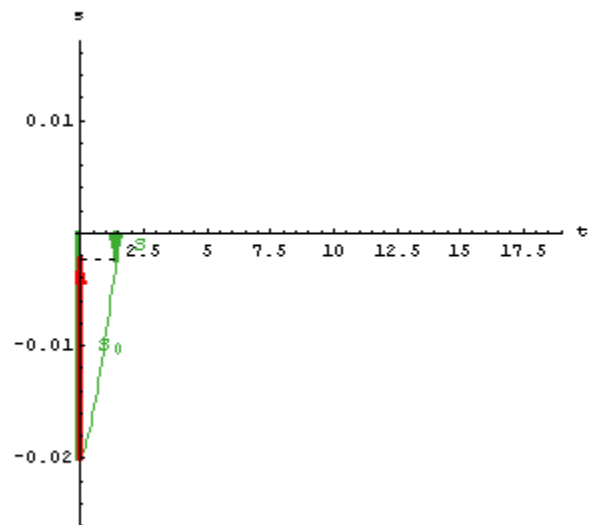
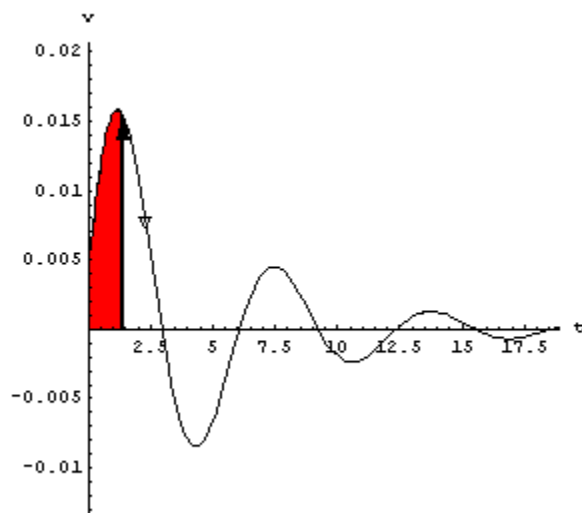
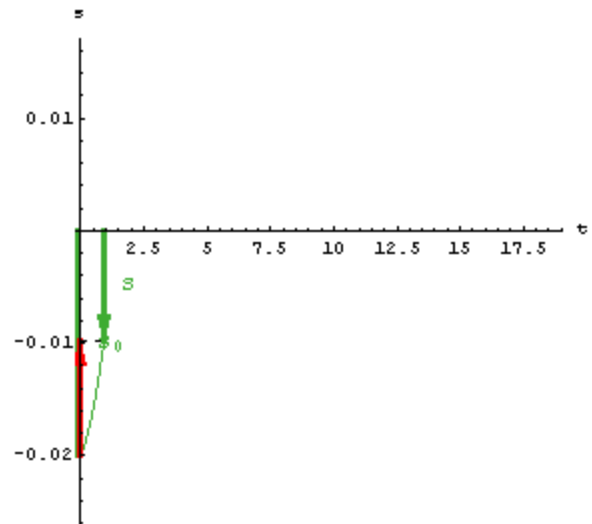
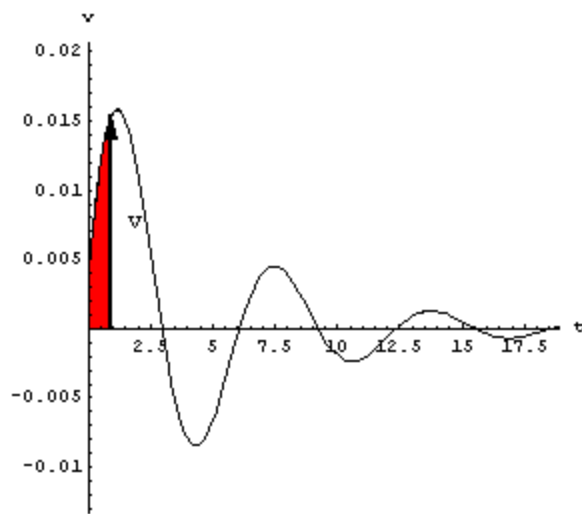


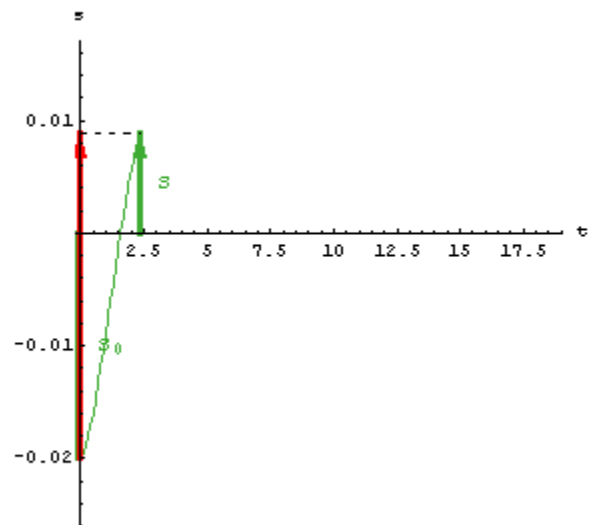
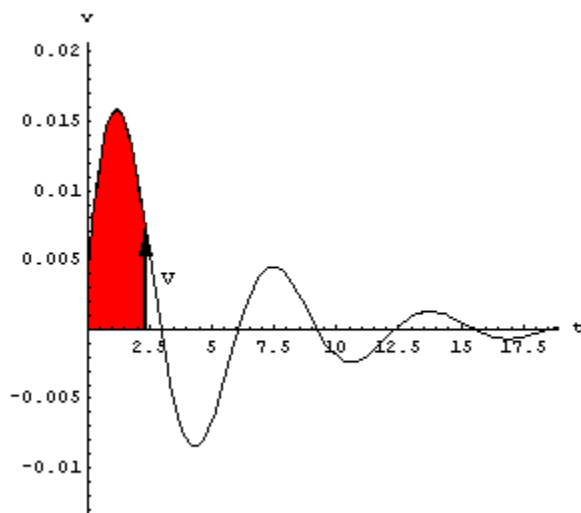
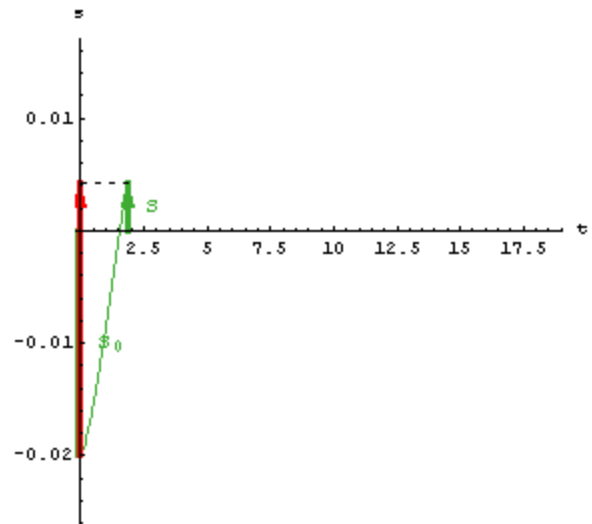
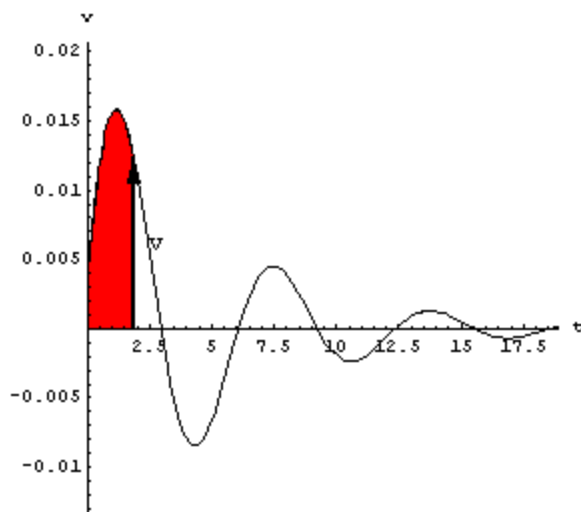
3. Use the **velocity[]** and **position[]** commands to visualize the integral relationships between the acceleration and velocity and between the velocity and position.

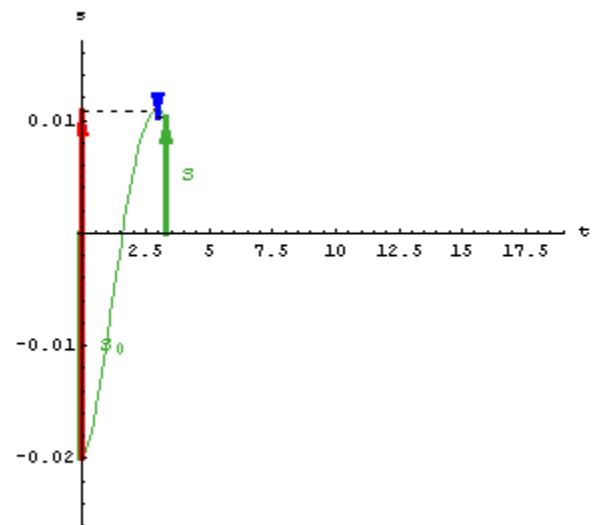
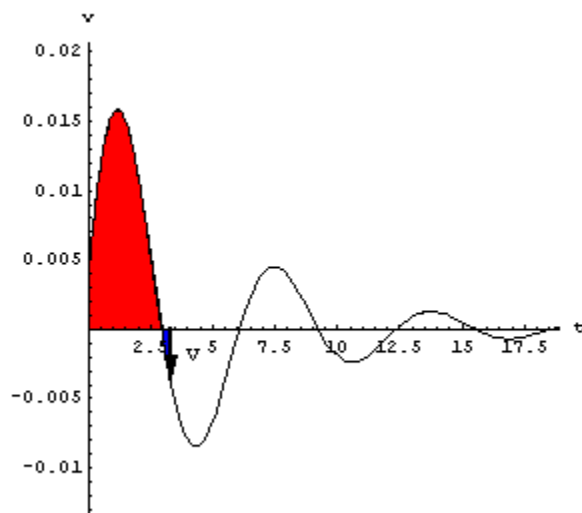
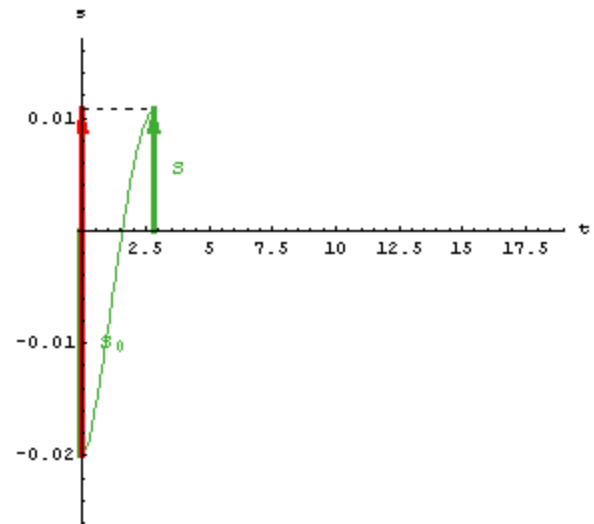
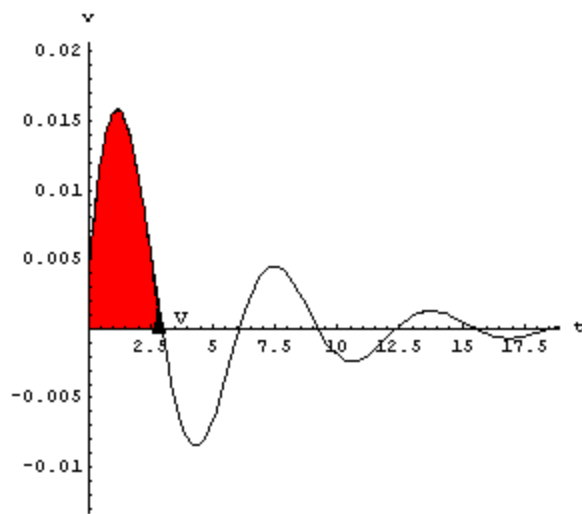
In[62]:=

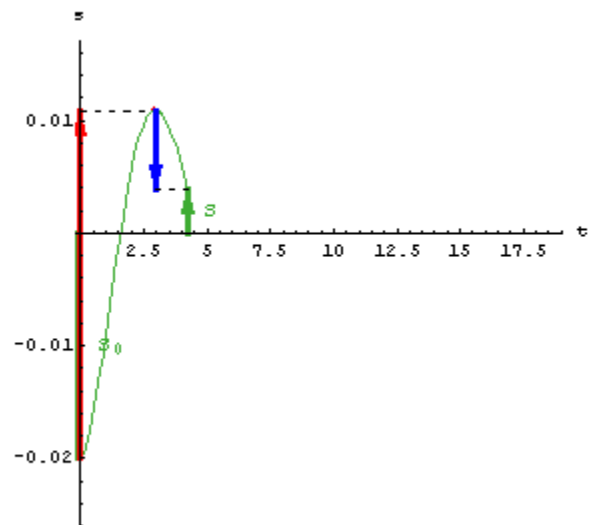
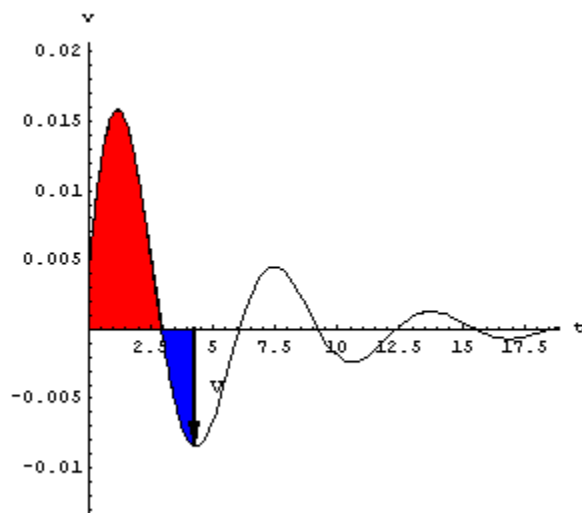
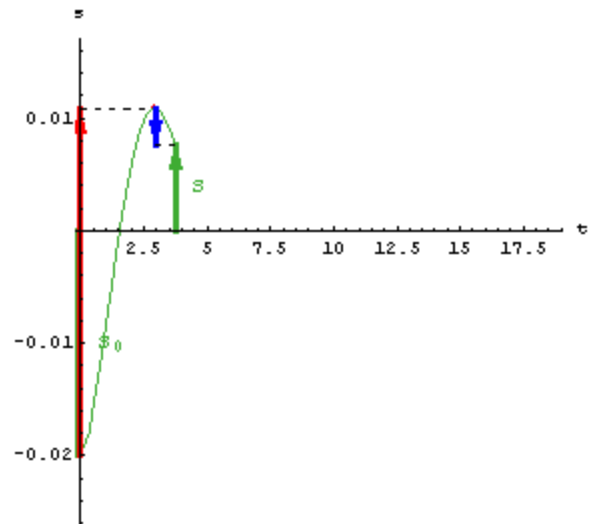
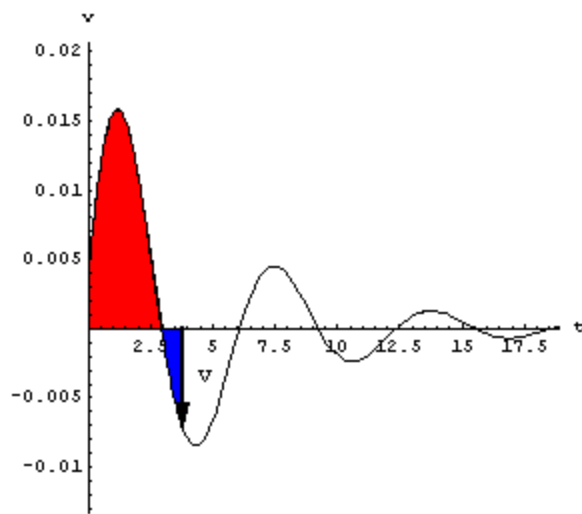
```
position[v[t], {t, 0, 6 * Pi}, s[0]]
```

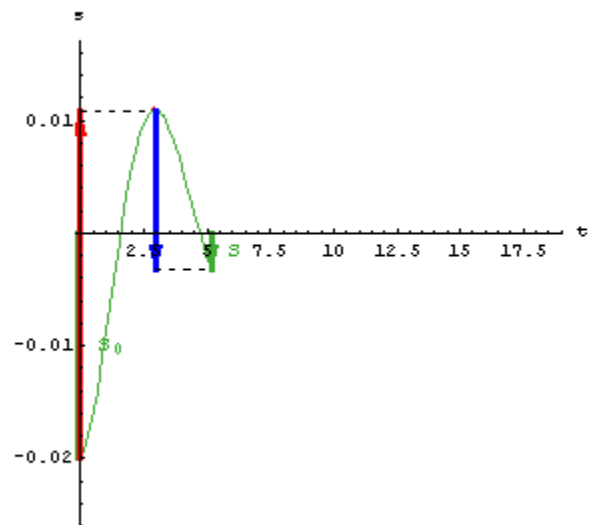
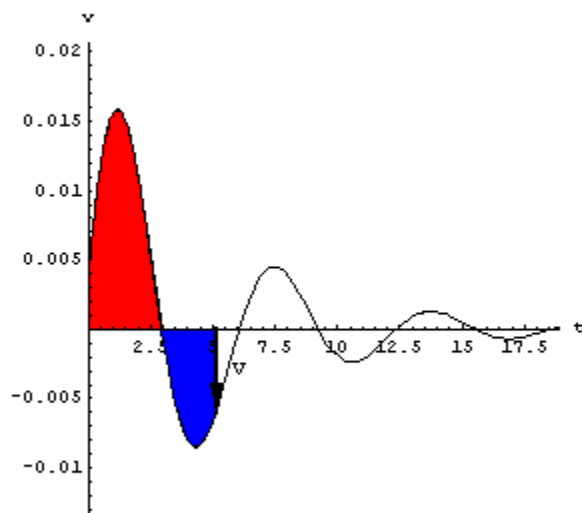
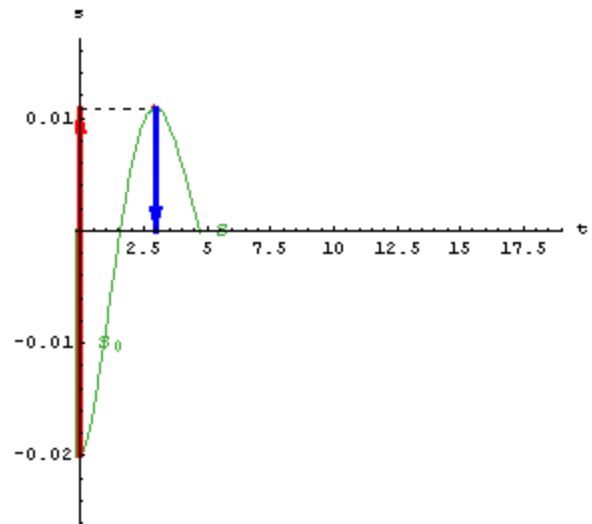
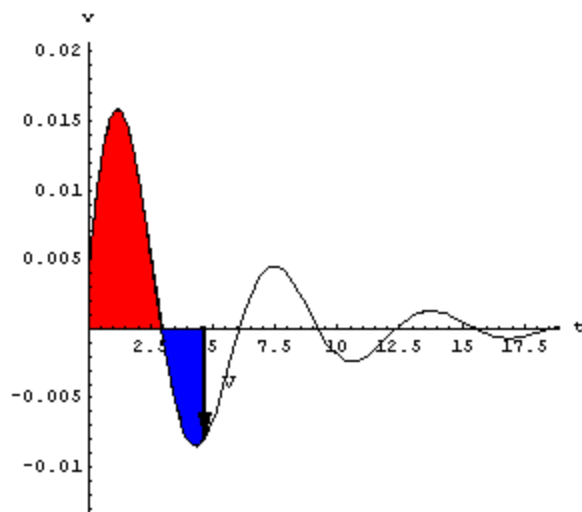


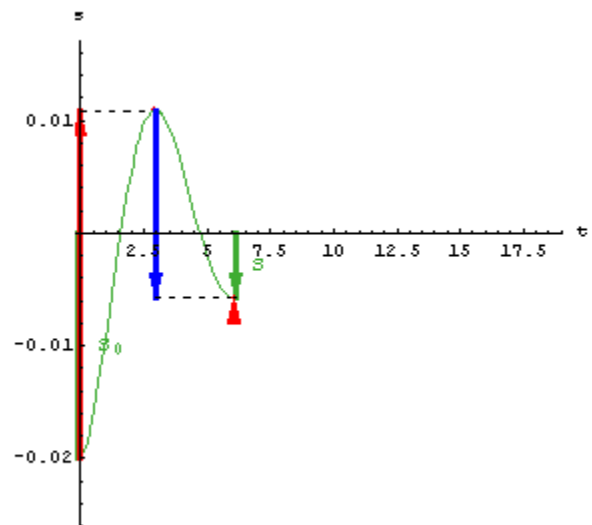
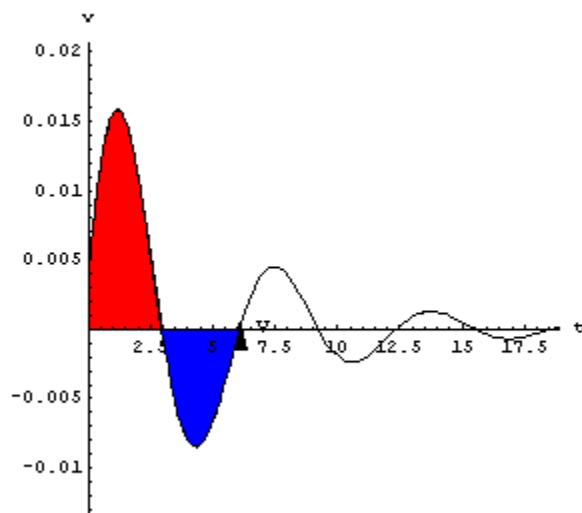
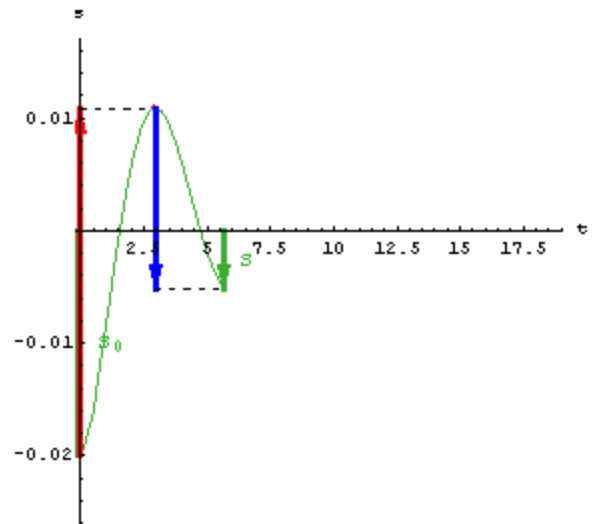
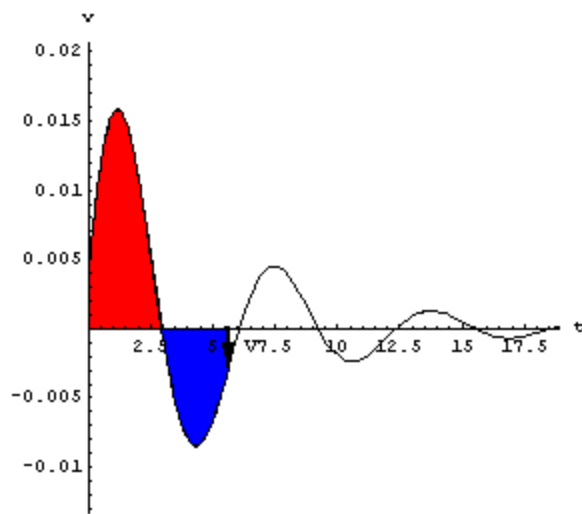


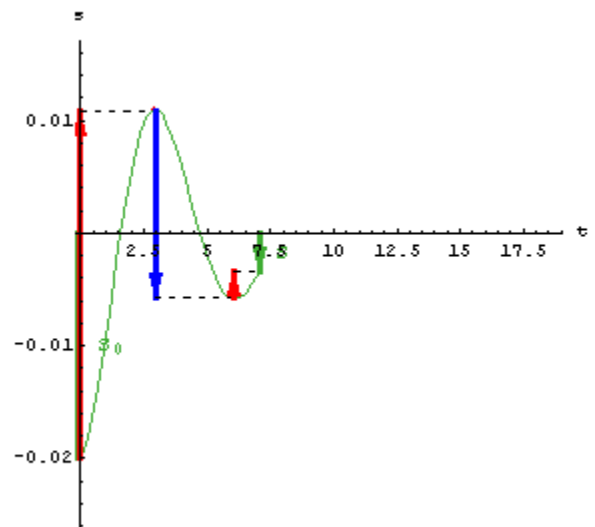
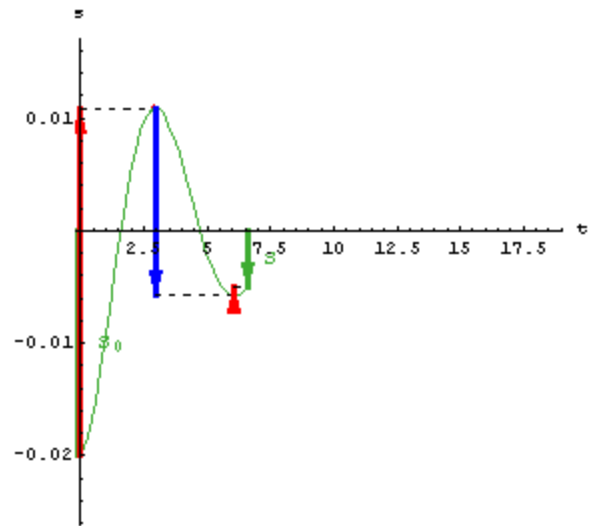


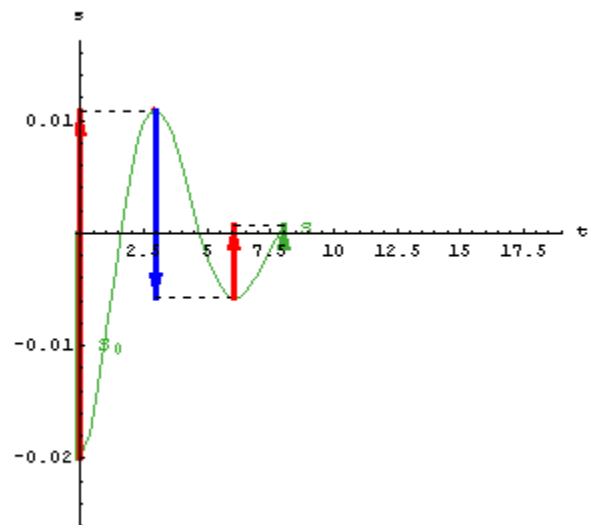
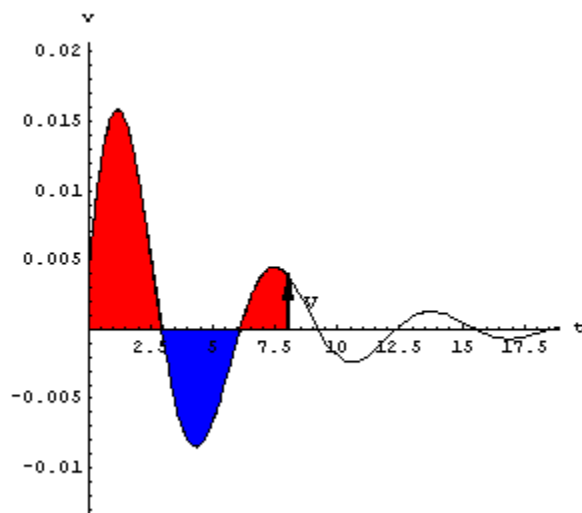
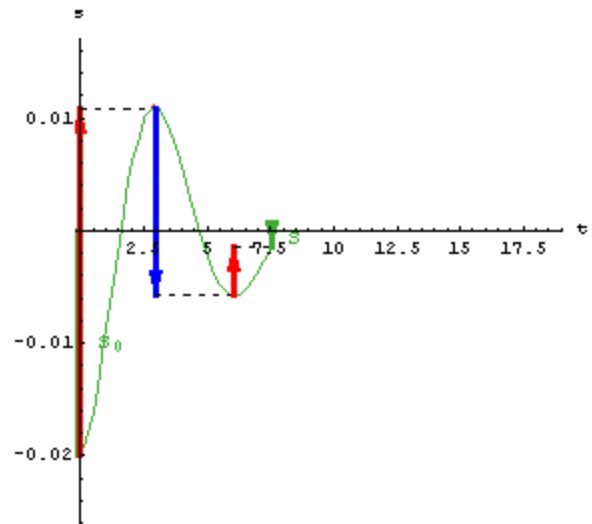
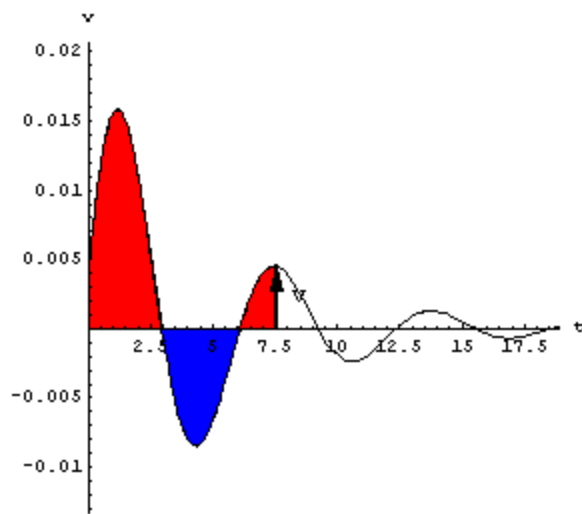


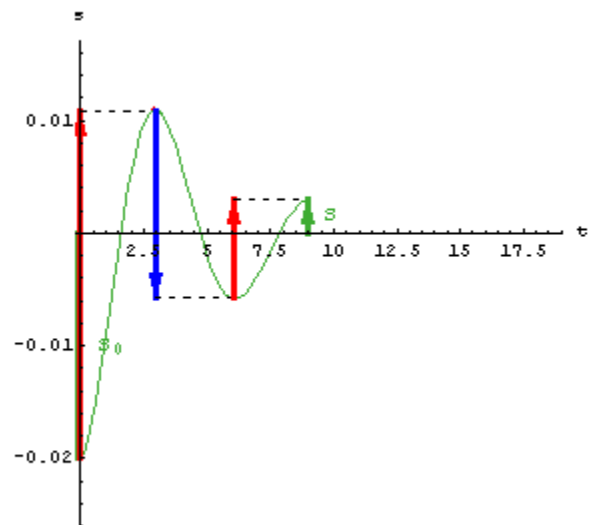
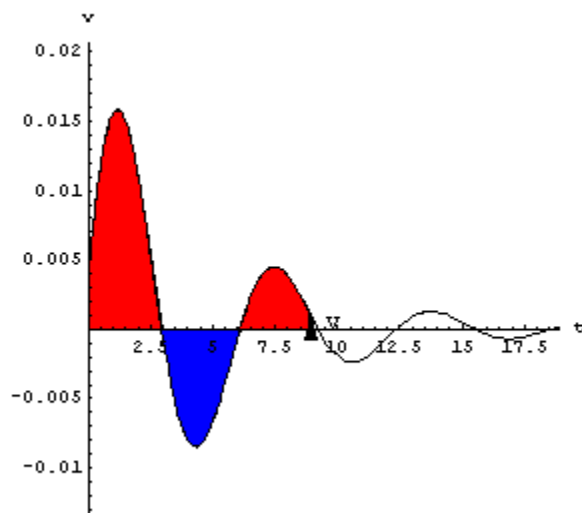
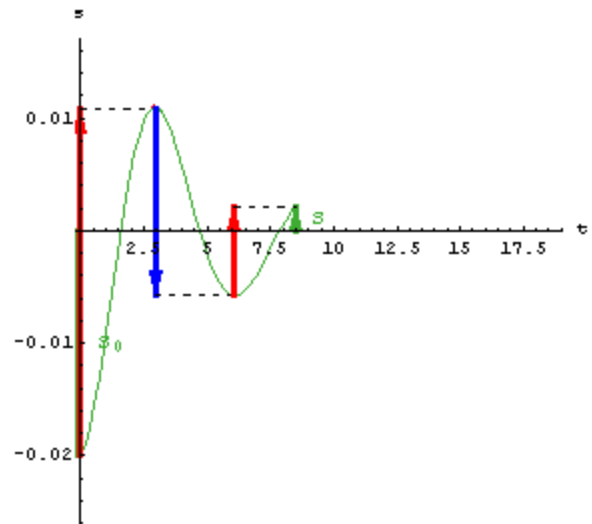
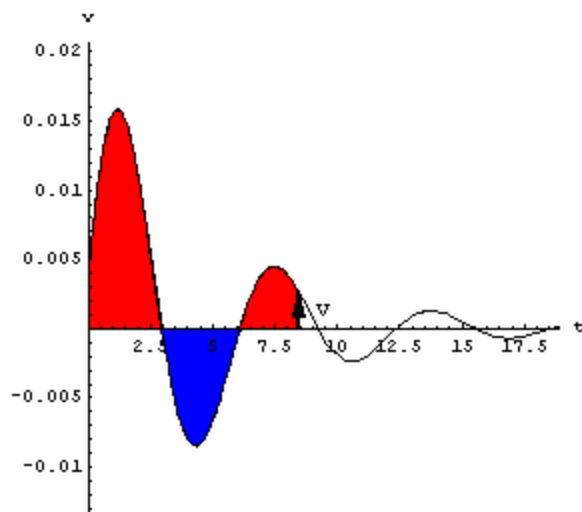


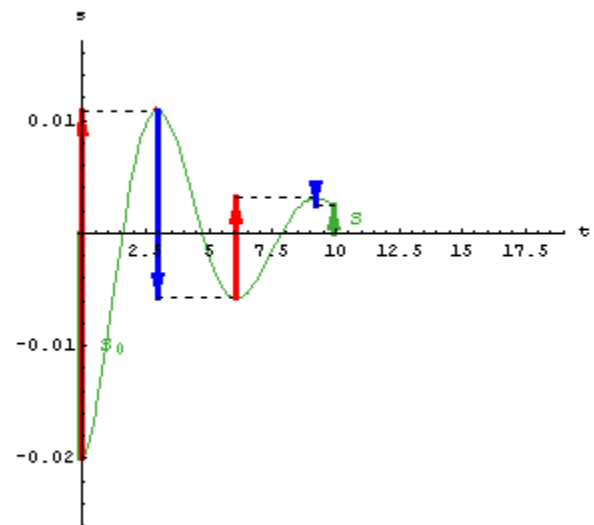
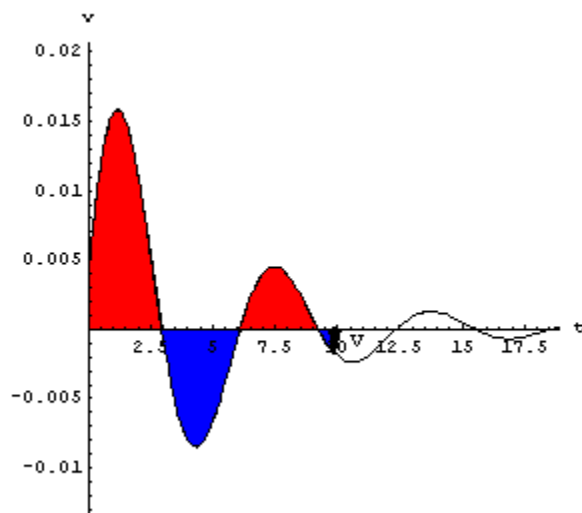
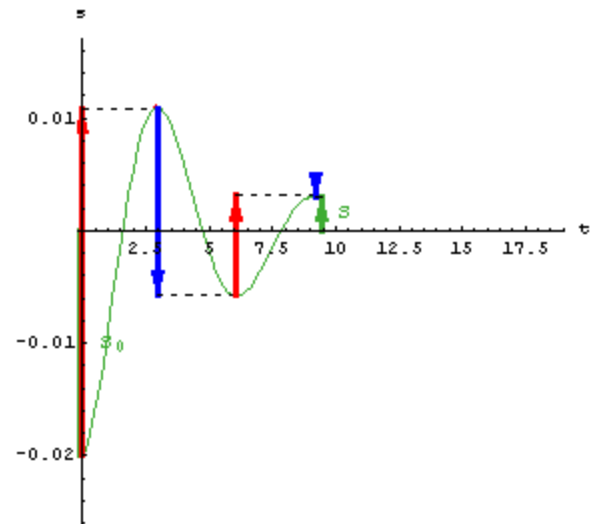
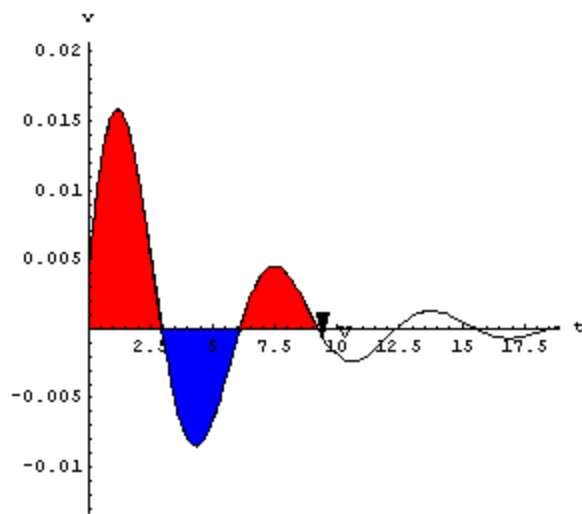


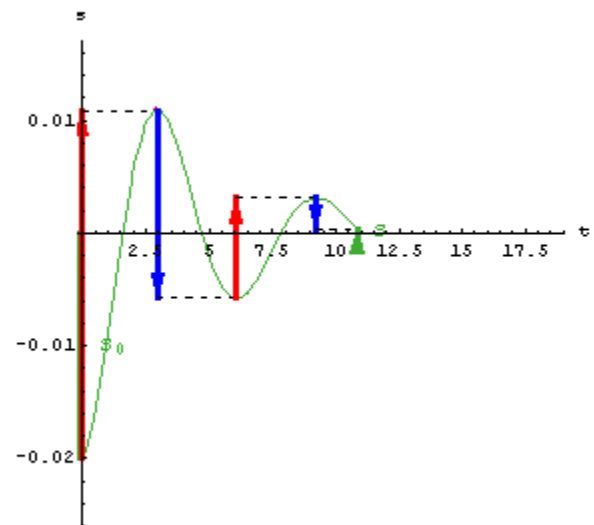
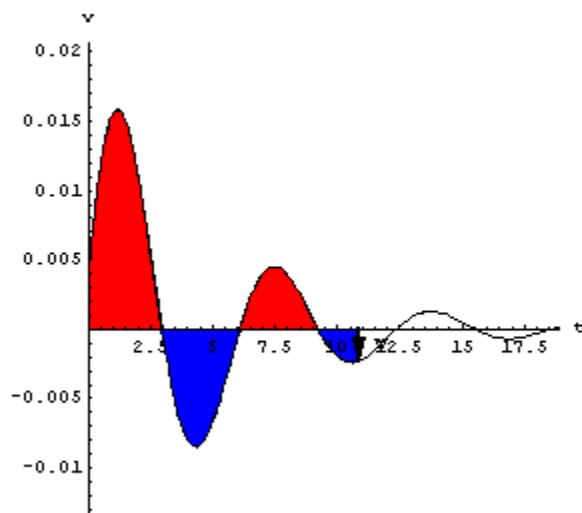
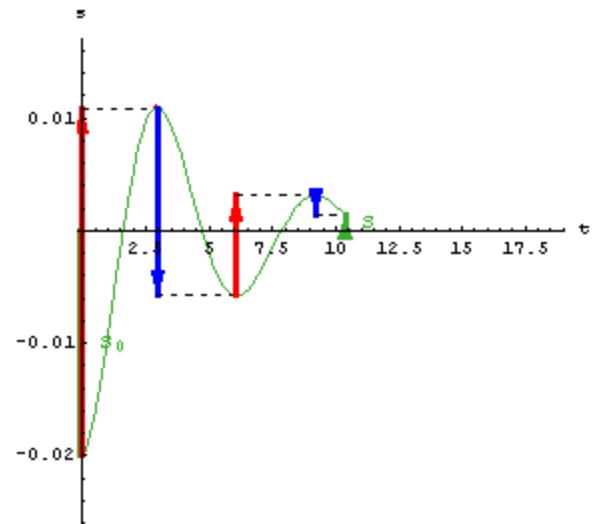
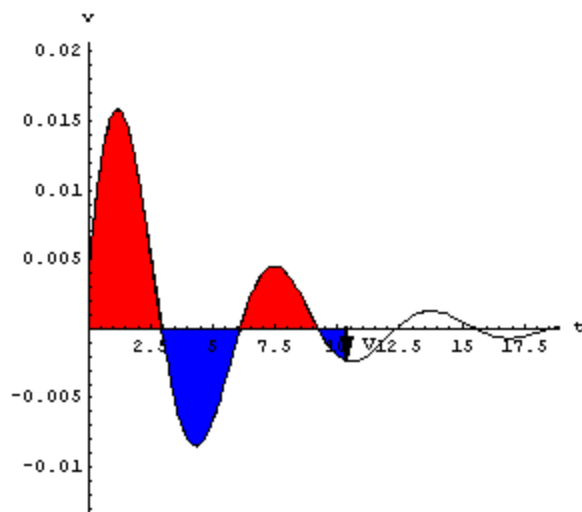


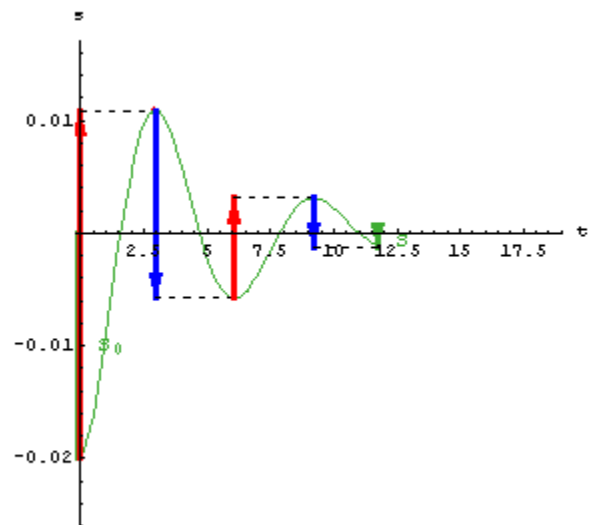
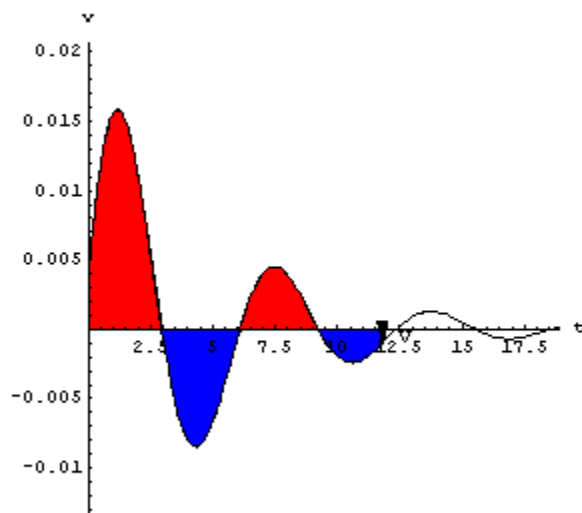
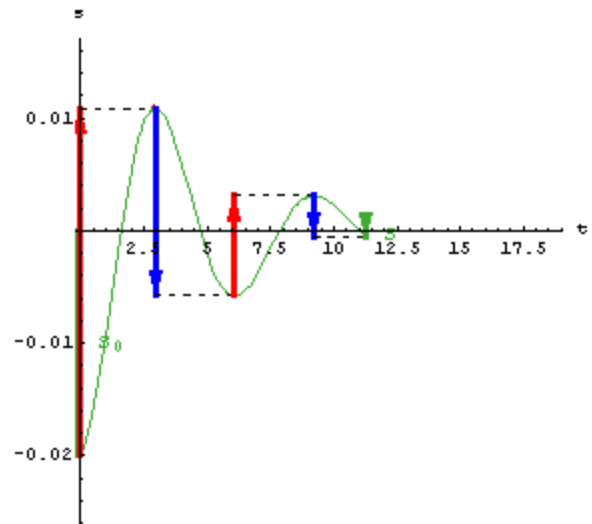
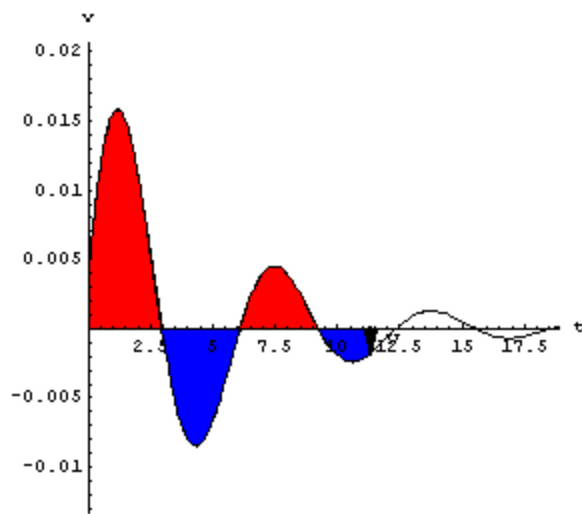


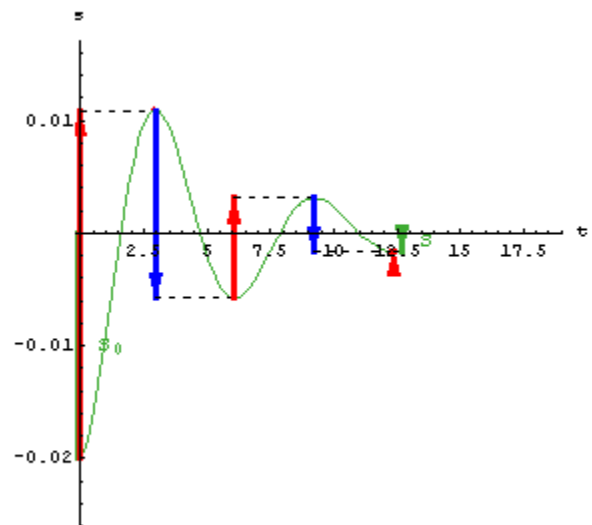
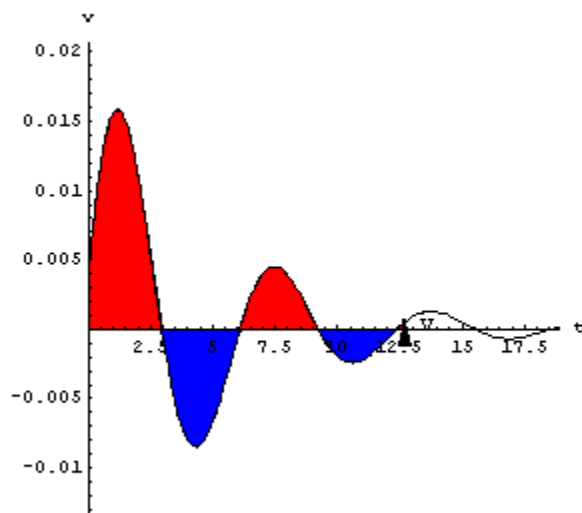
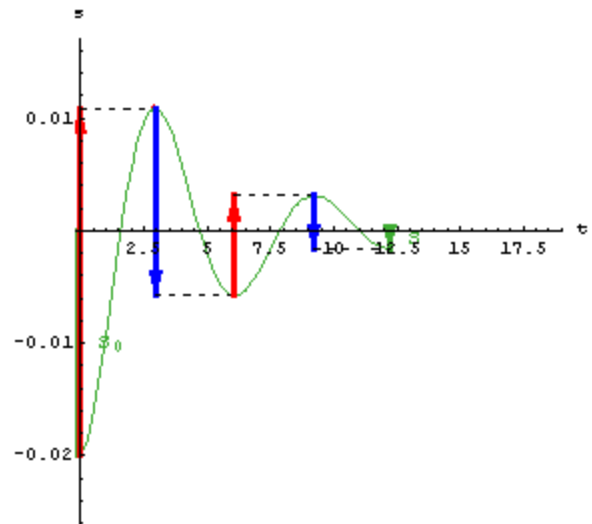
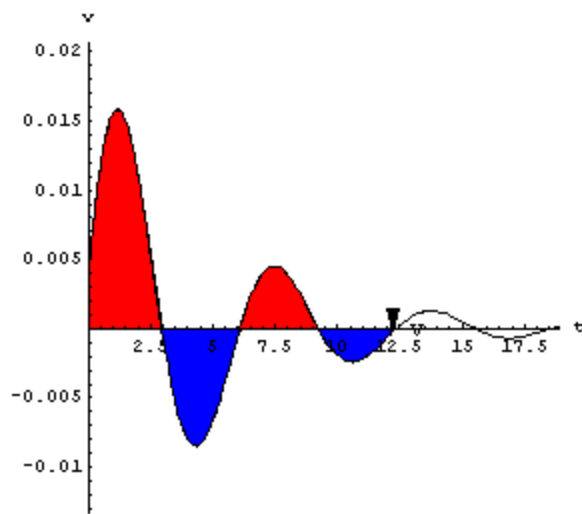


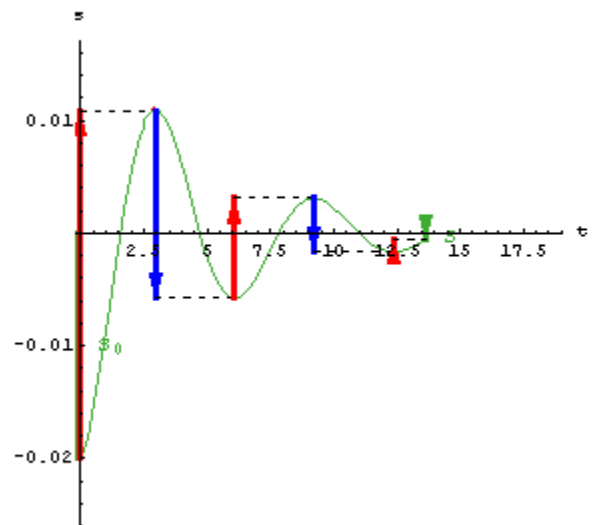
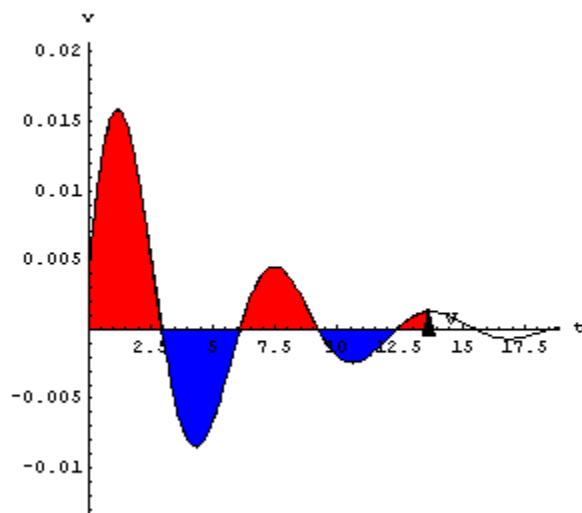
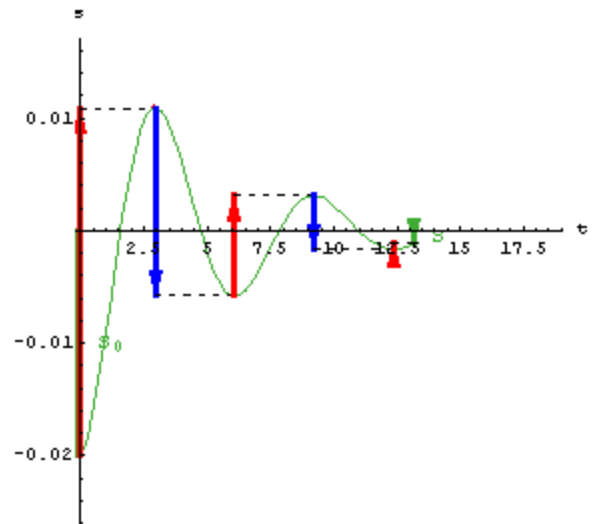
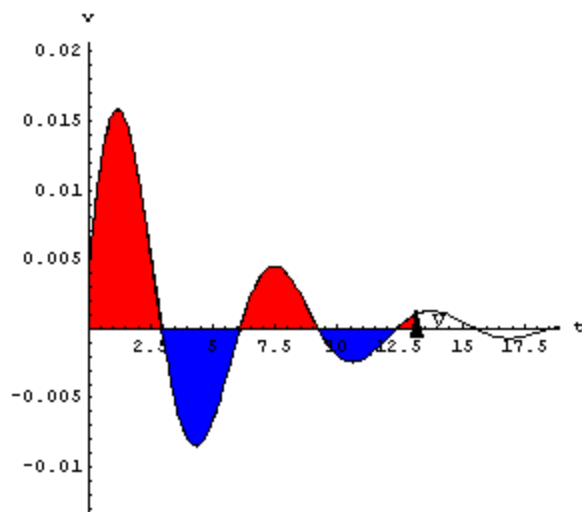


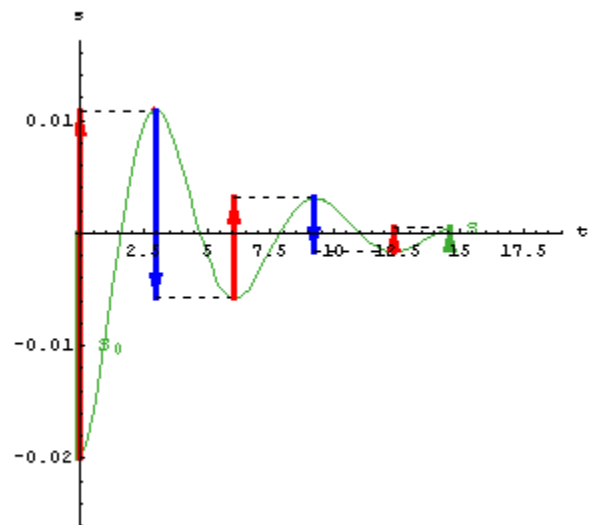
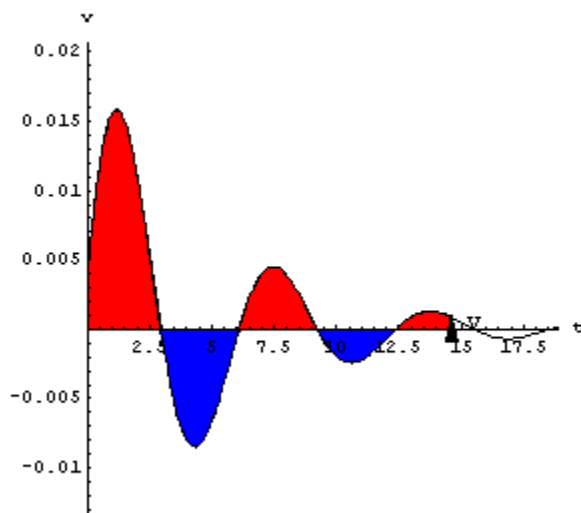
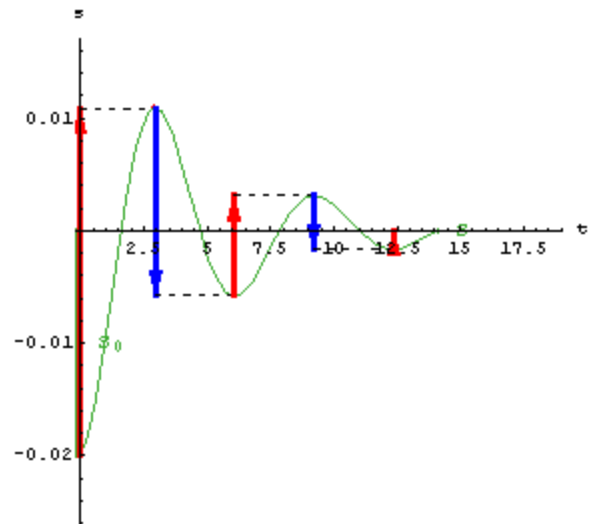
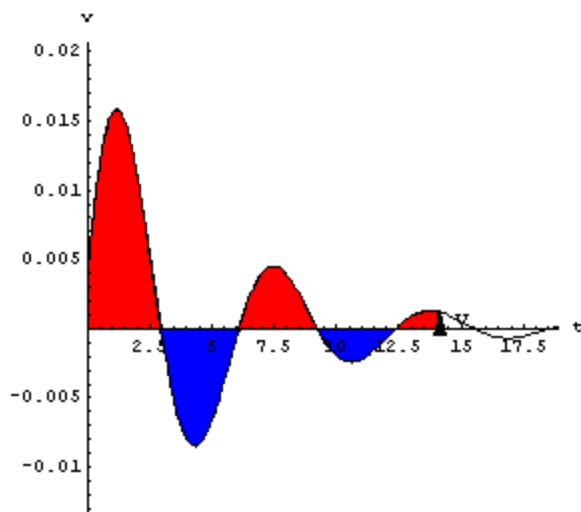


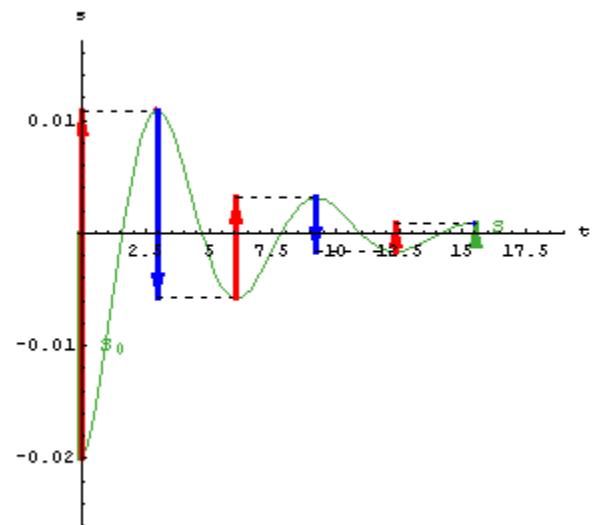
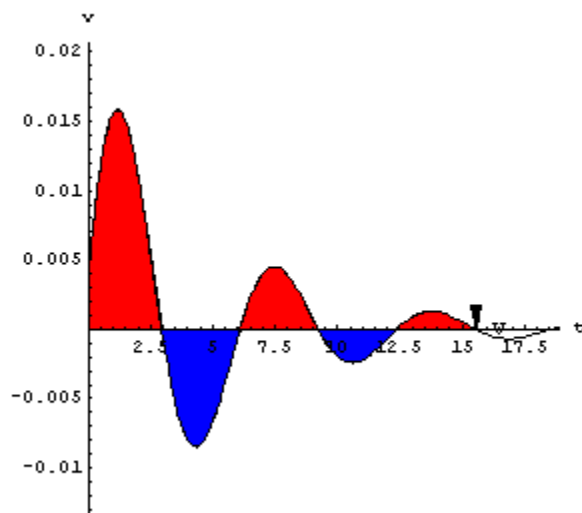
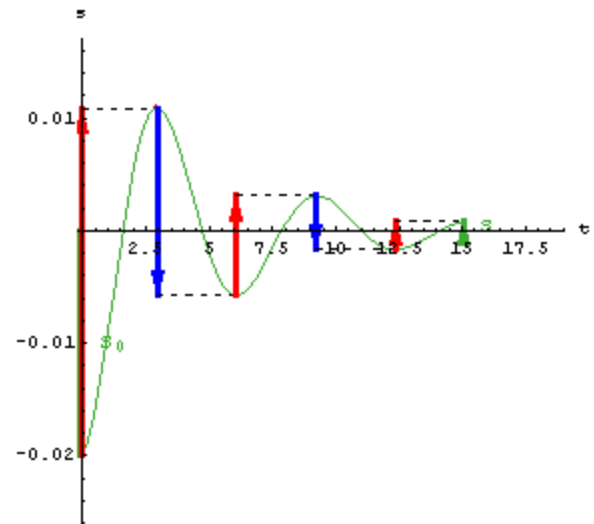
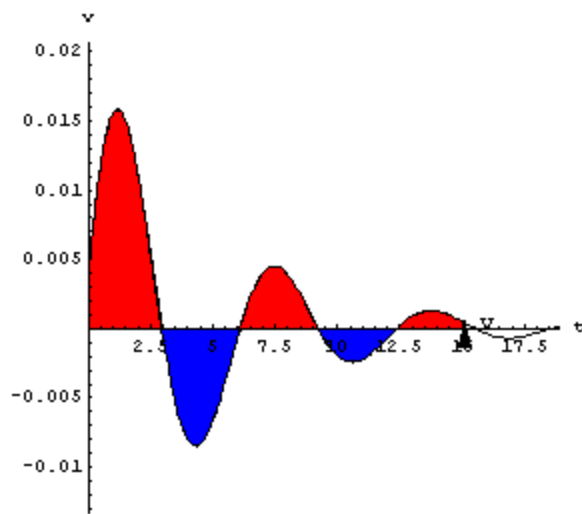


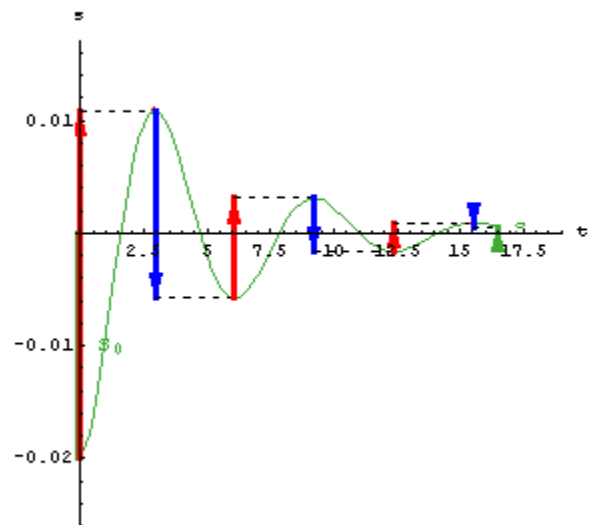
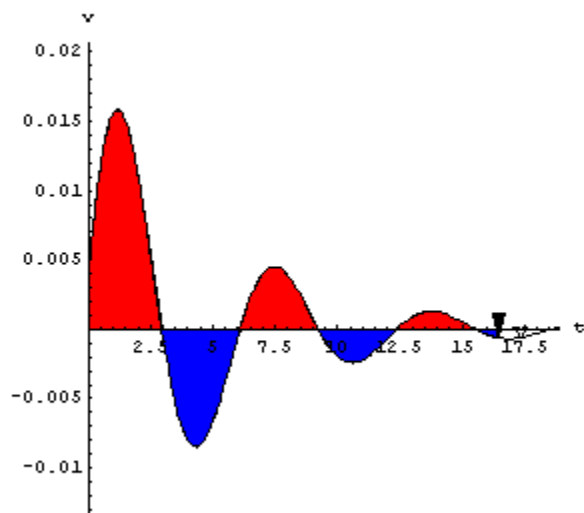
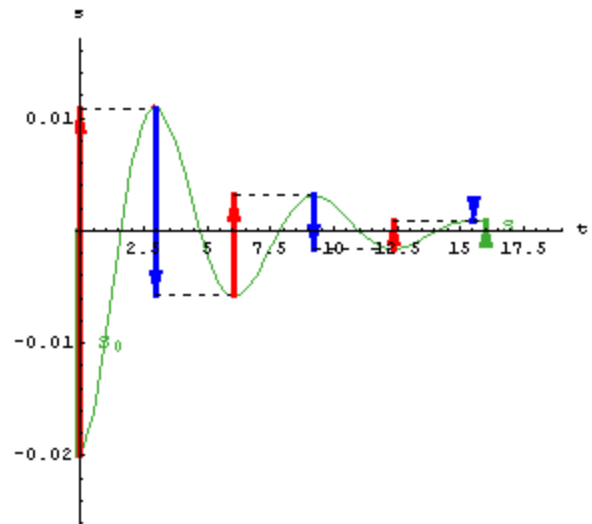
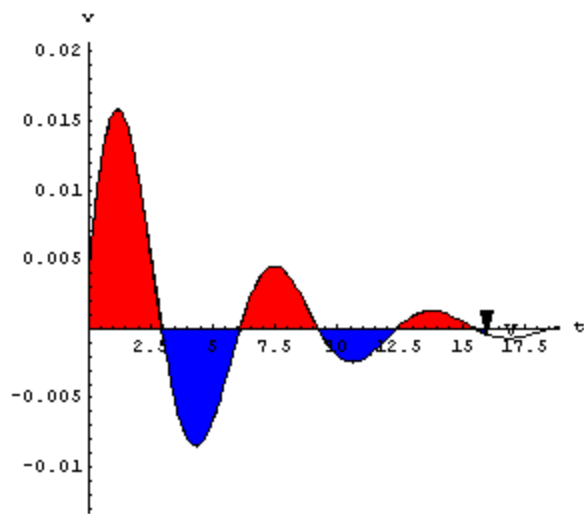


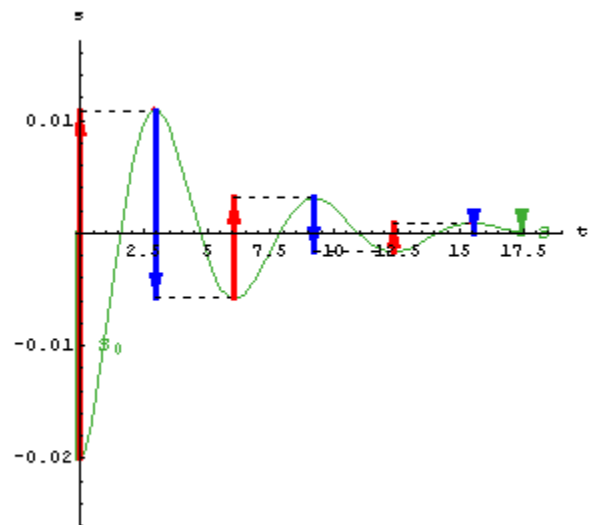
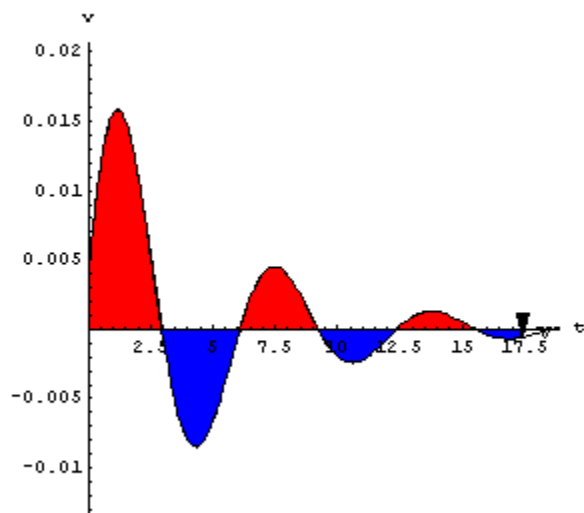
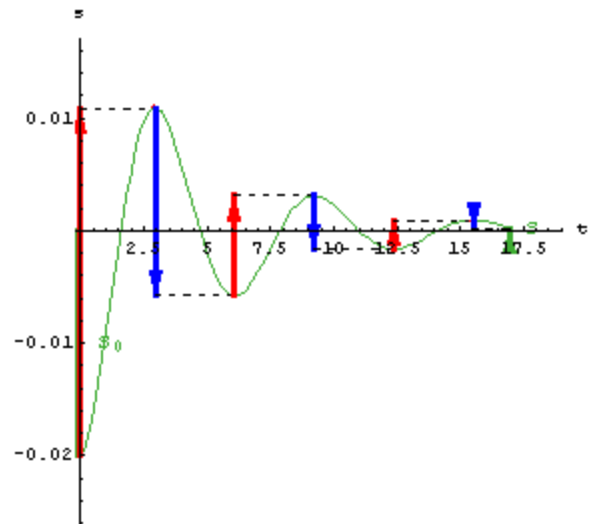
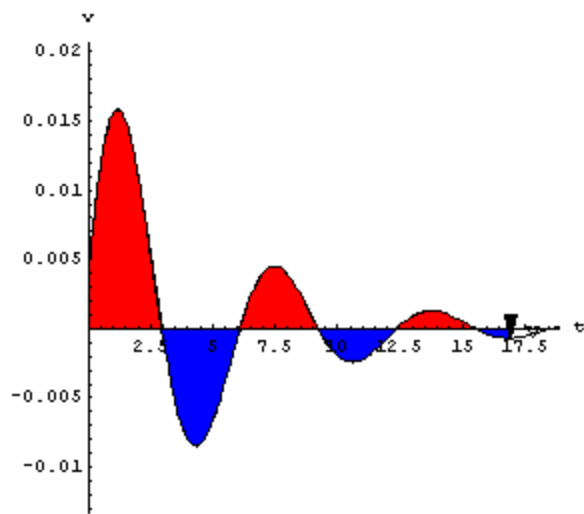


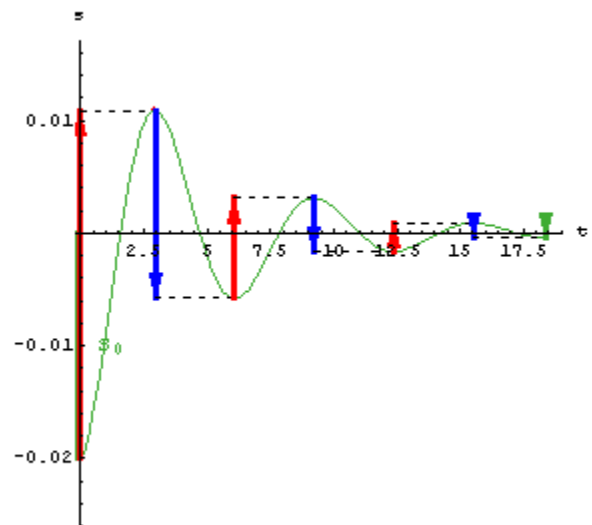
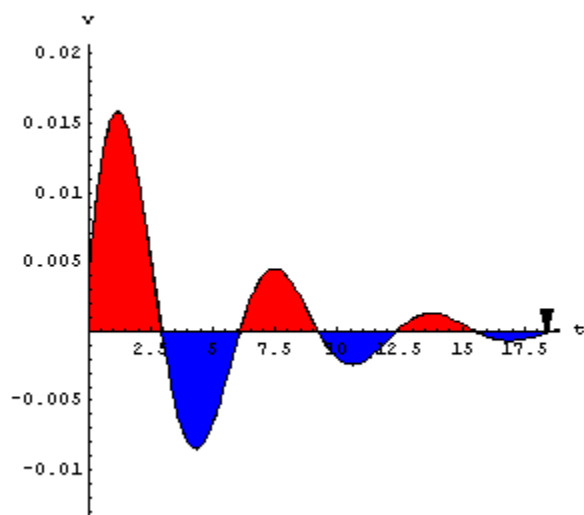
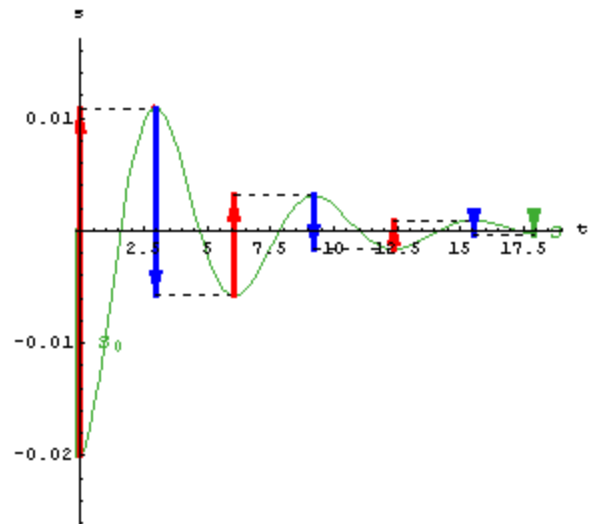
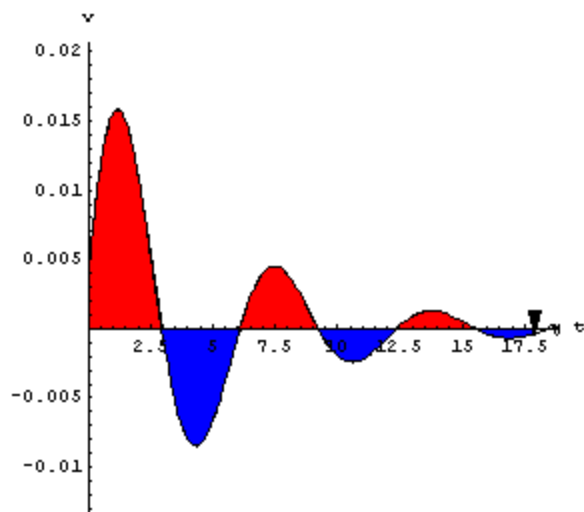


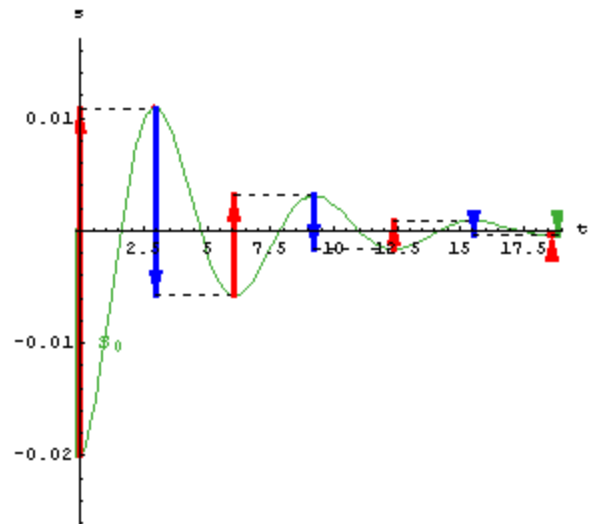
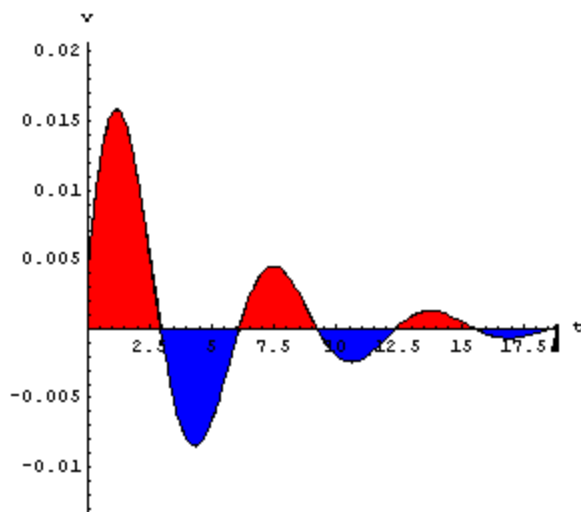






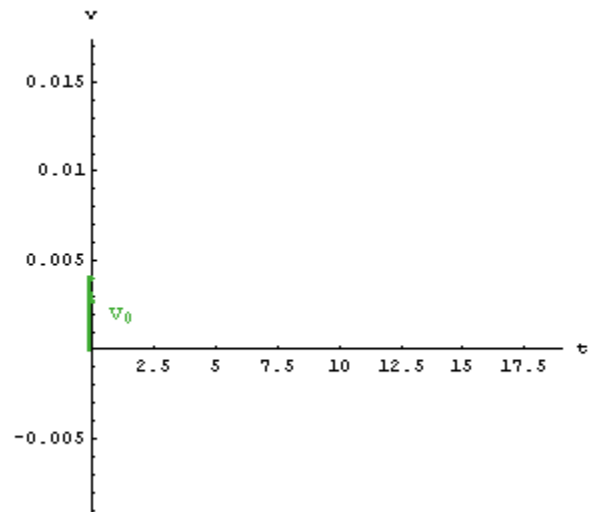
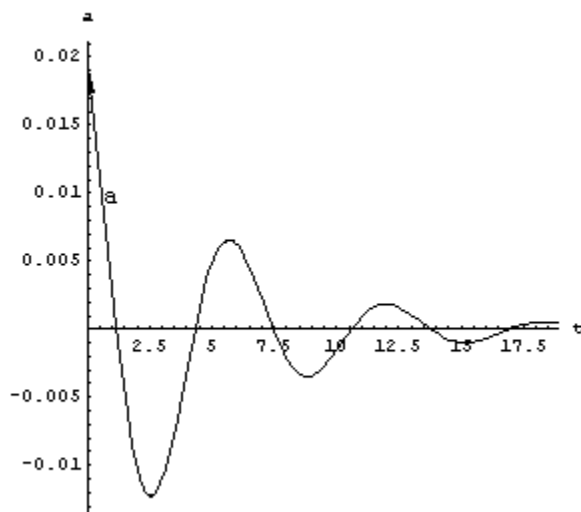


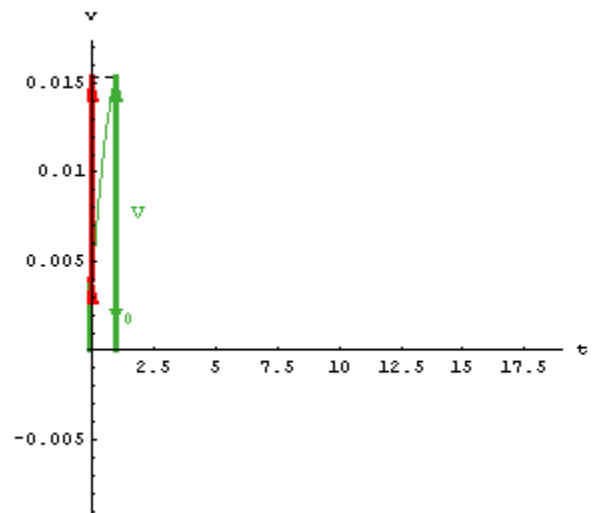
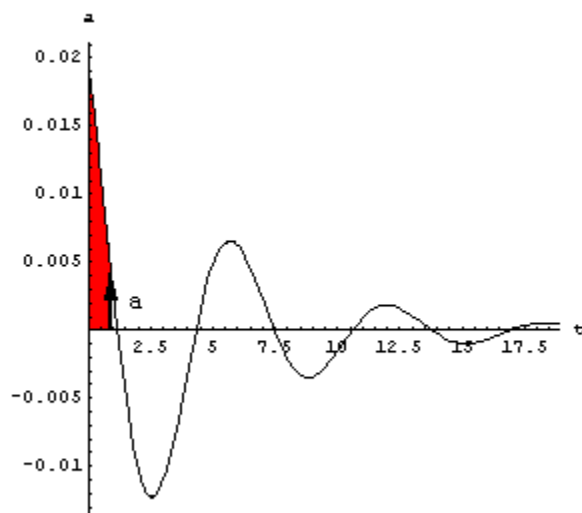
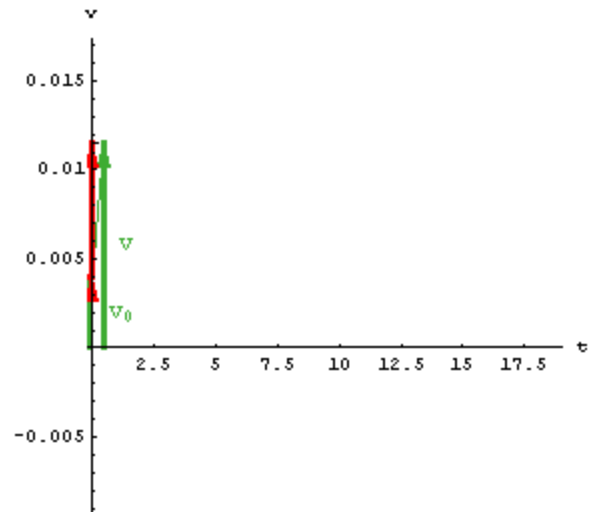
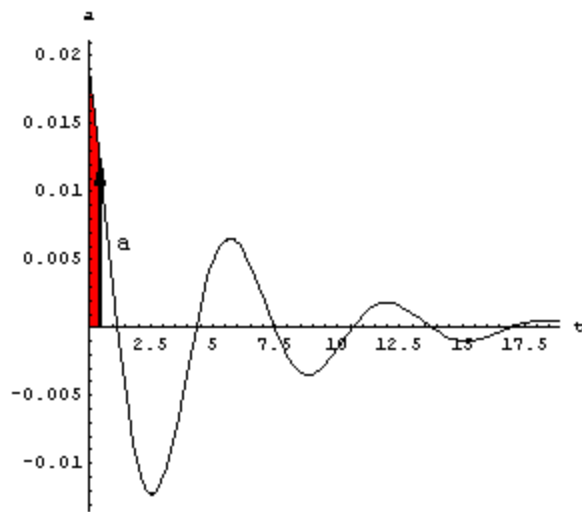


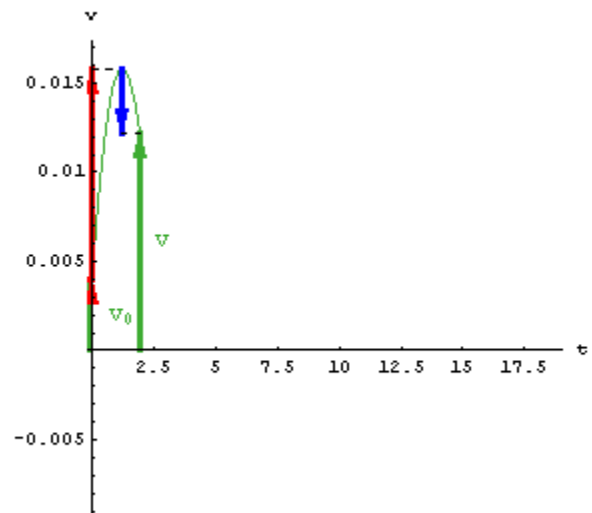
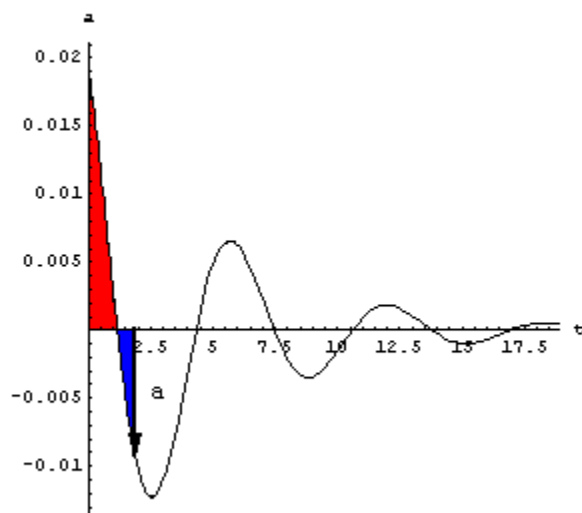
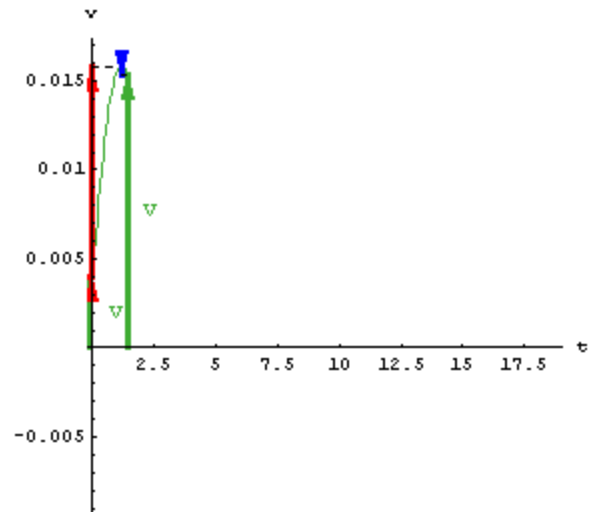
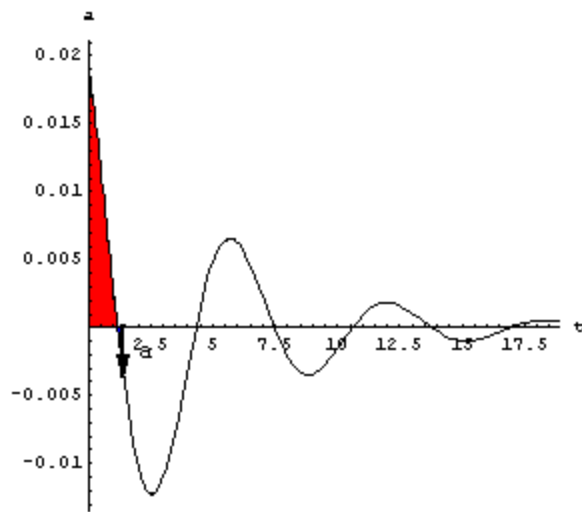


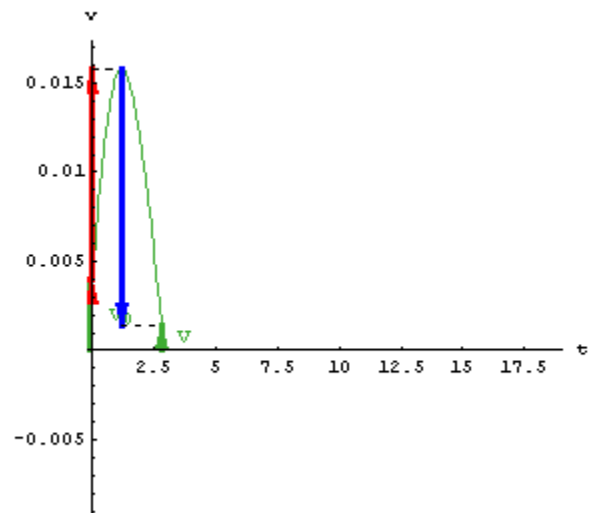
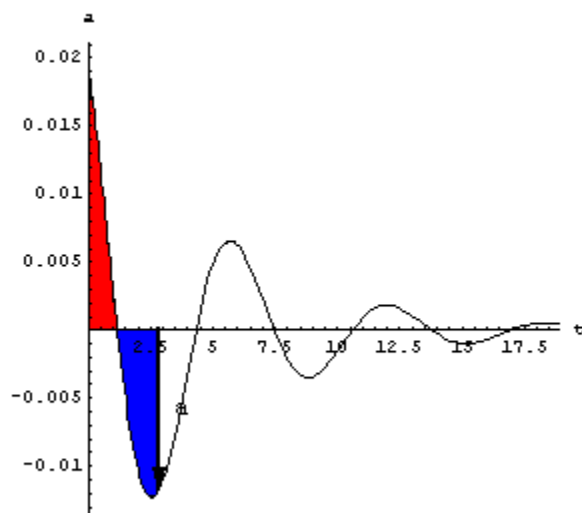
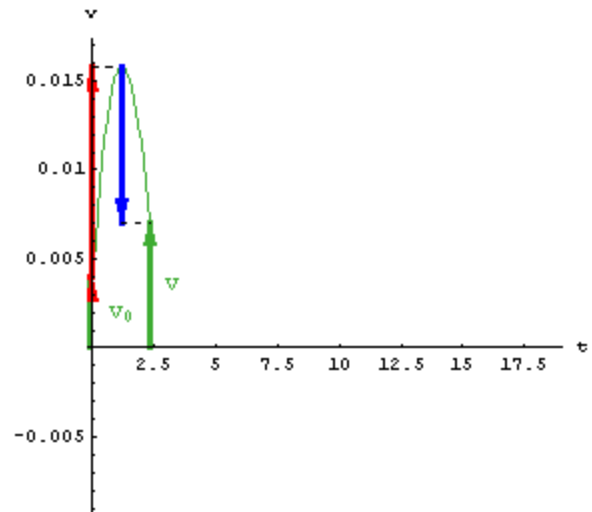
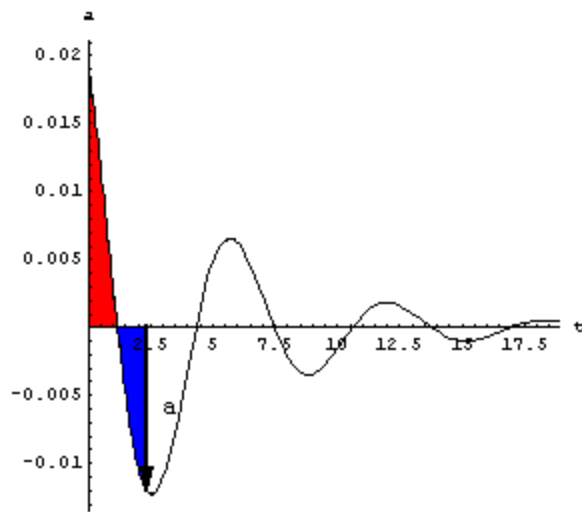
In[63]:=

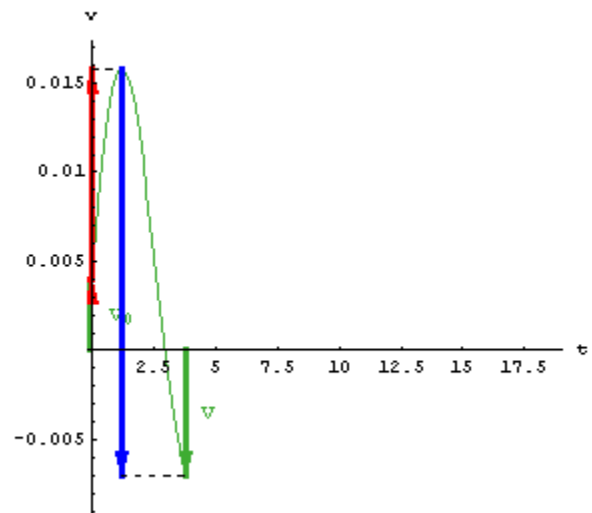
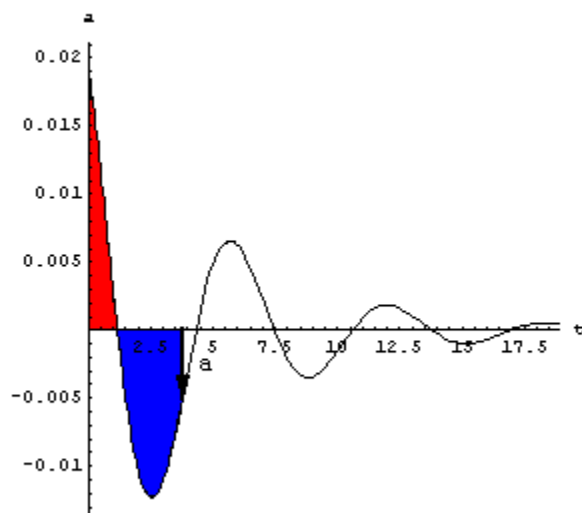
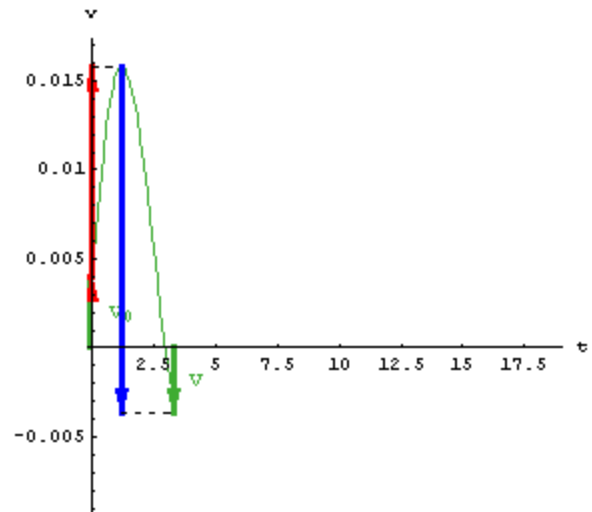
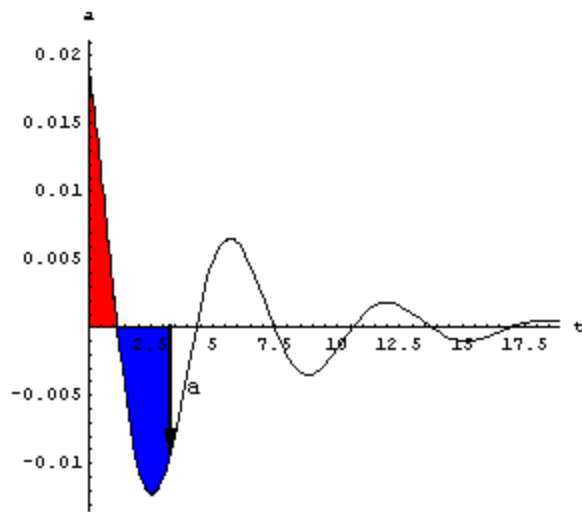
```
velocity[a[t], {t, 0, 6 * Pi}, v[0]]
```

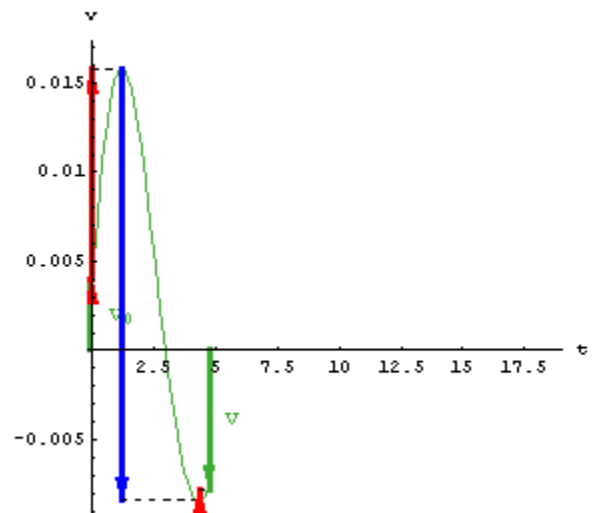
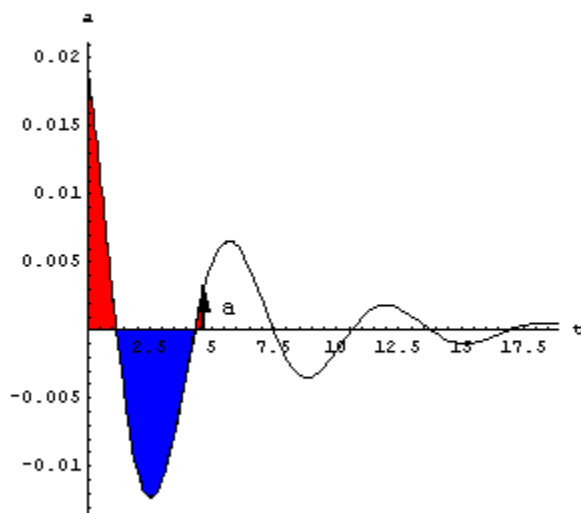
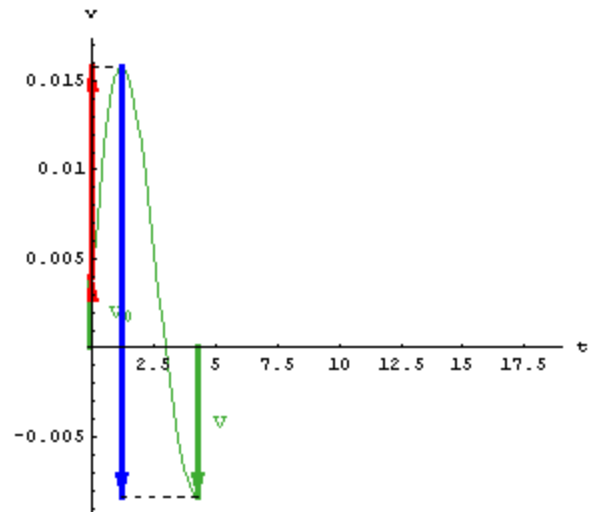
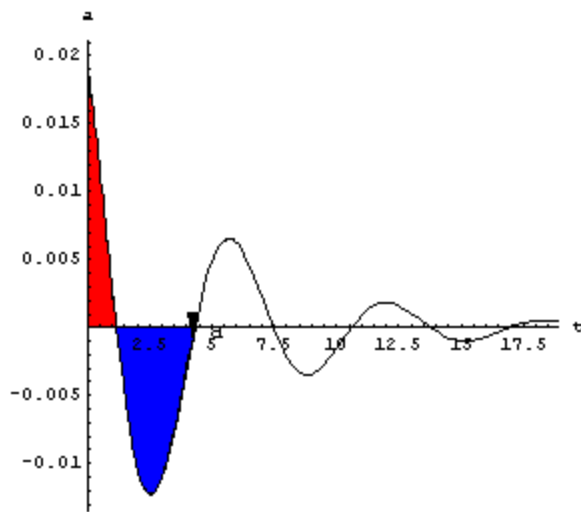


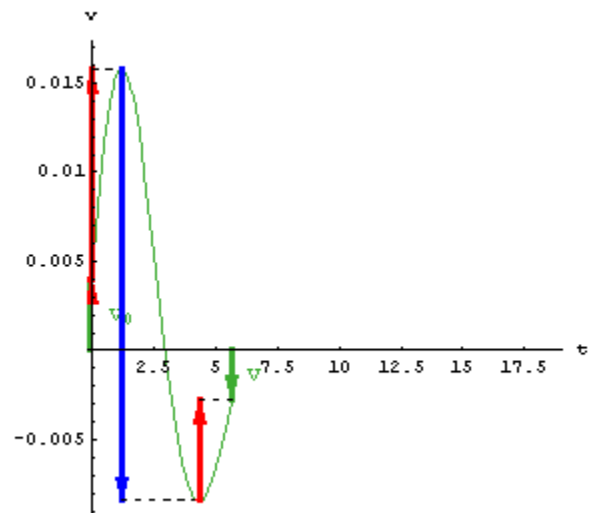
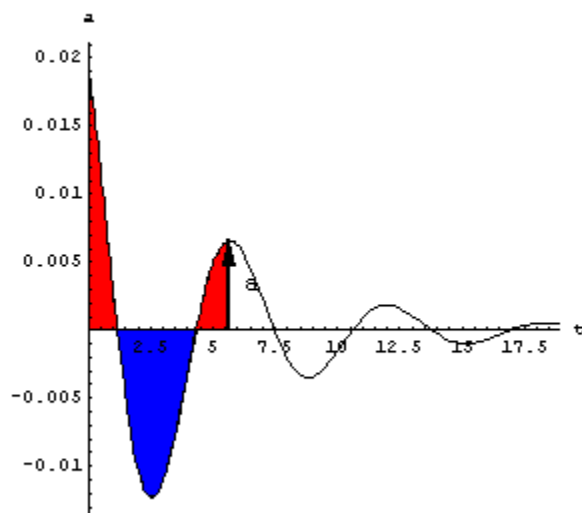
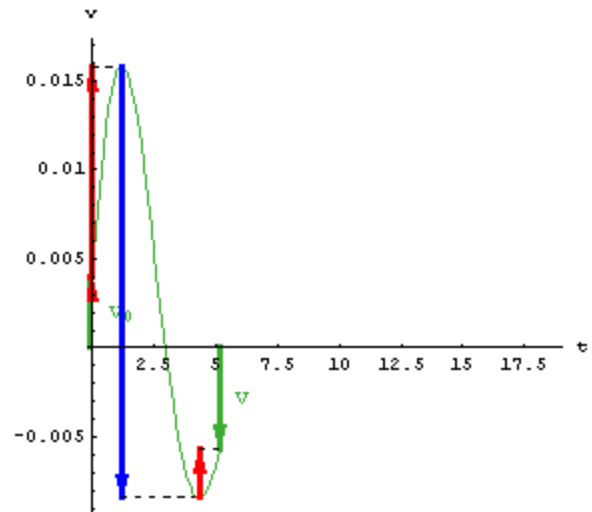
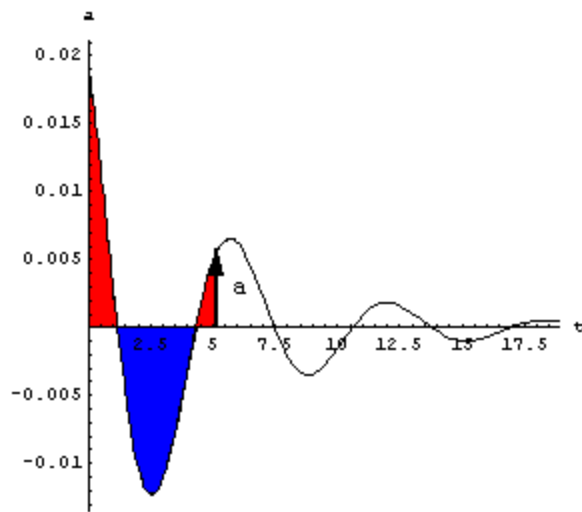


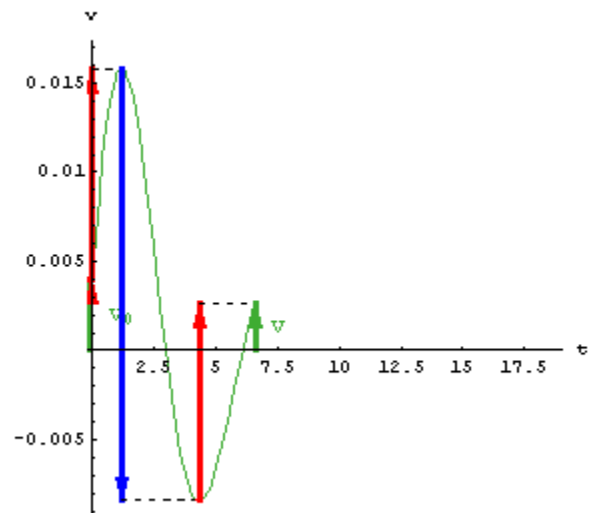
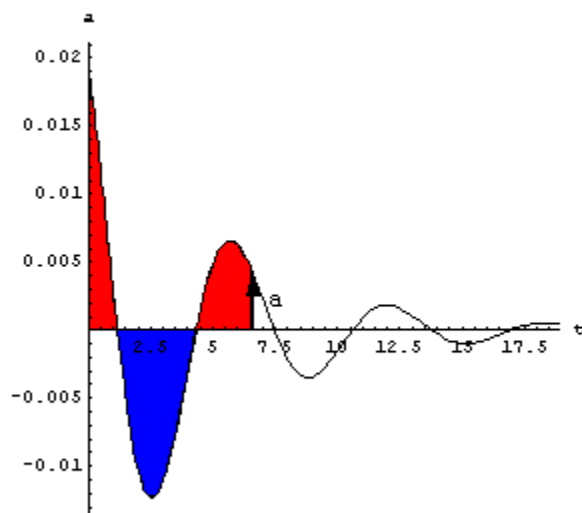
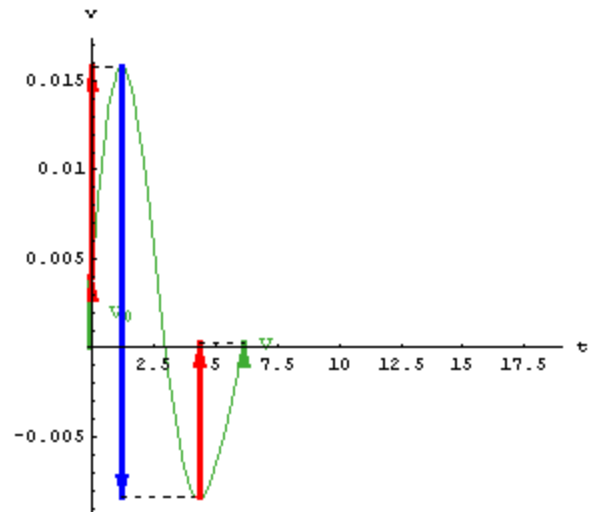
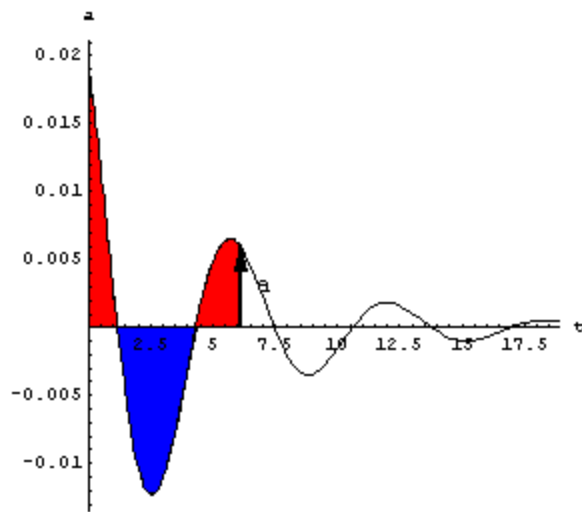


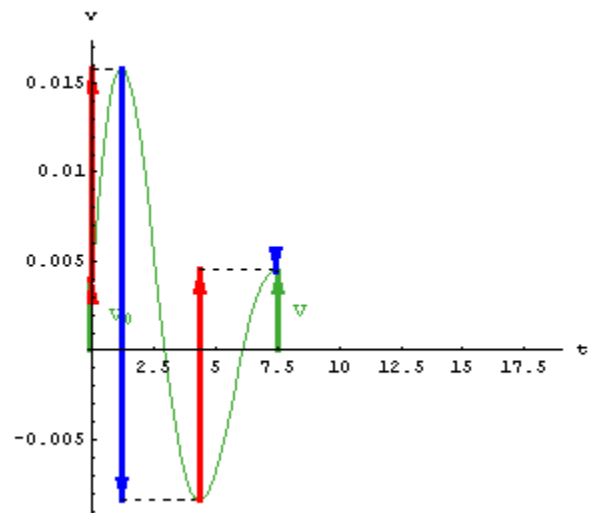
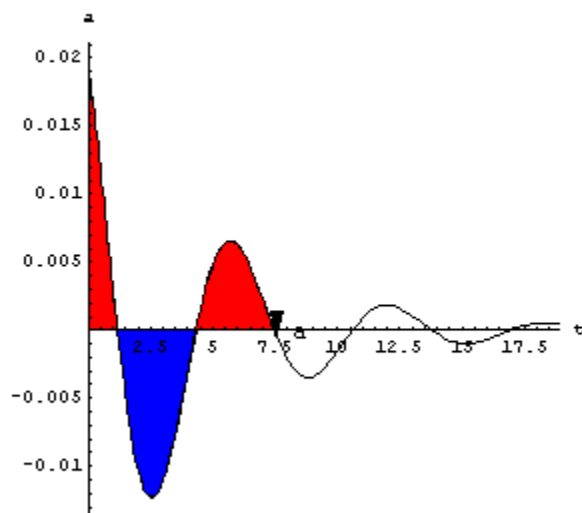
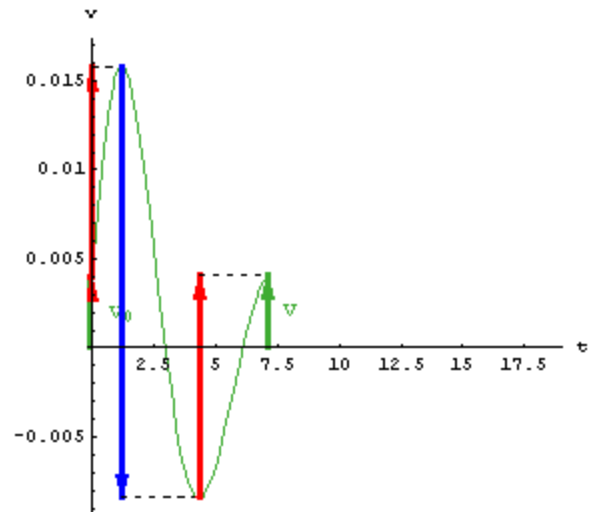
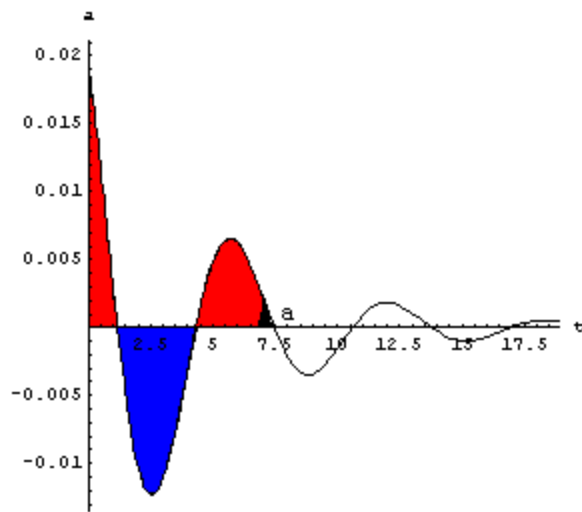


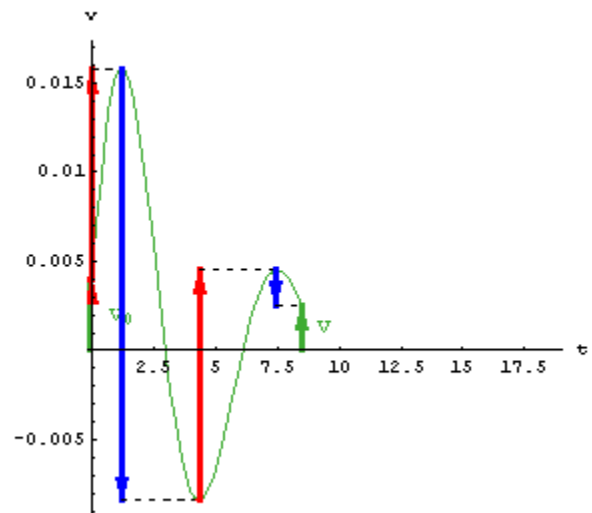
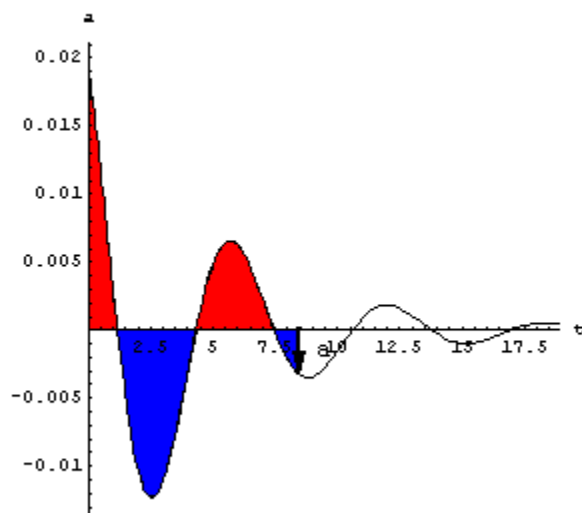
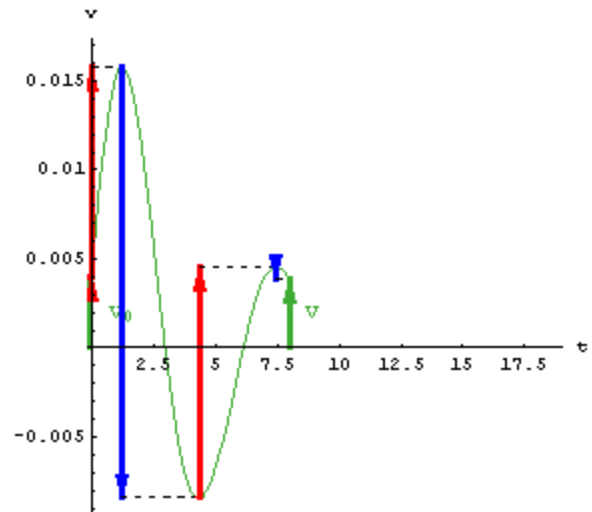
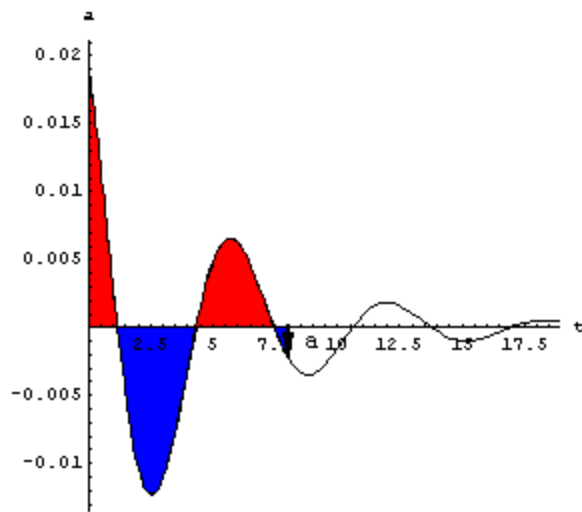


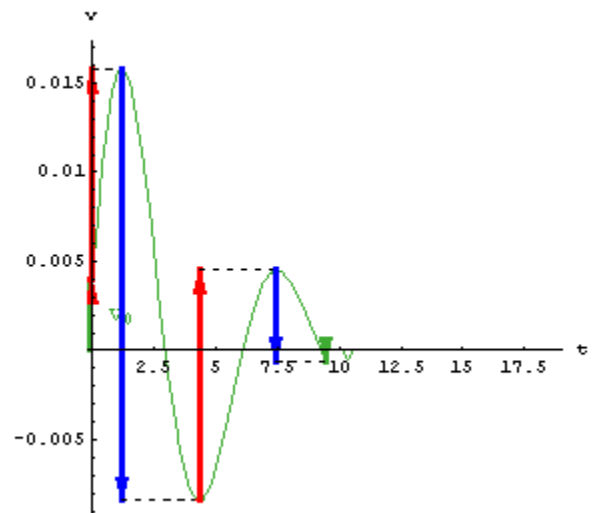
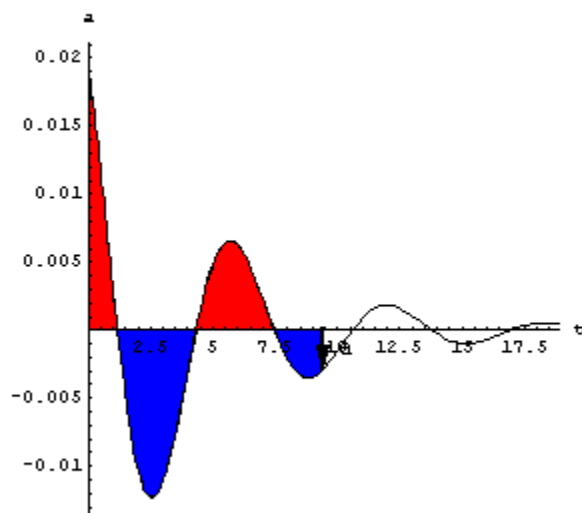
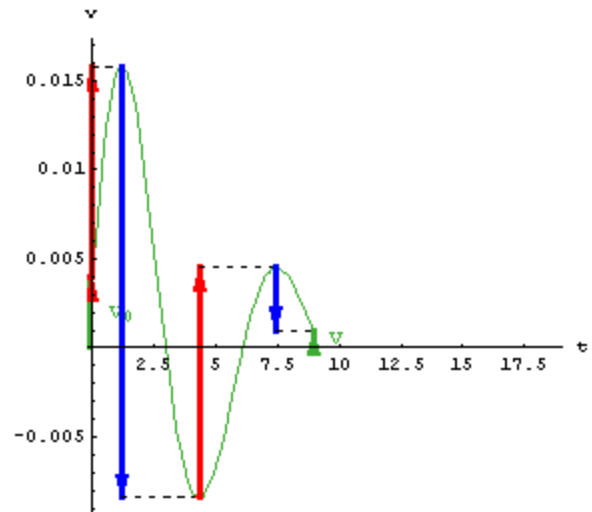
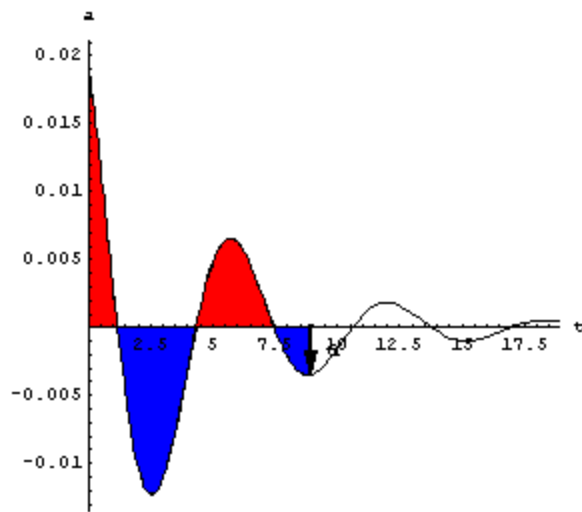


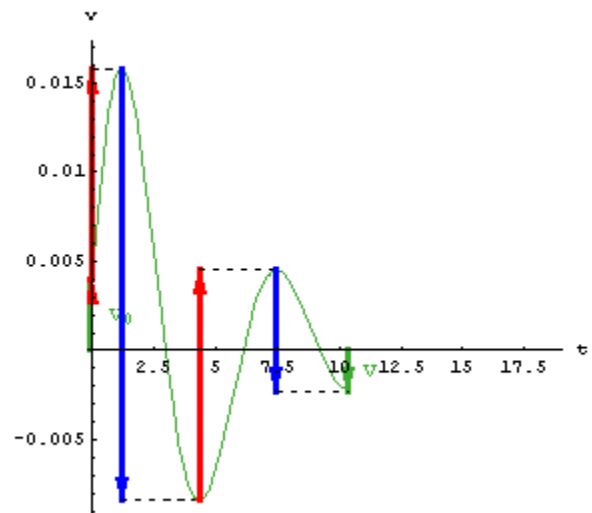
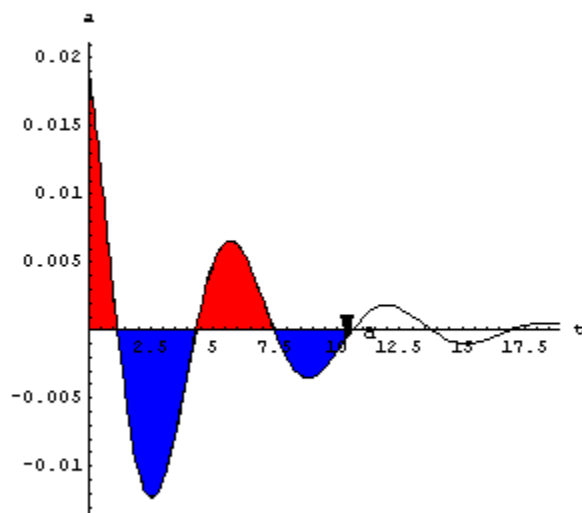
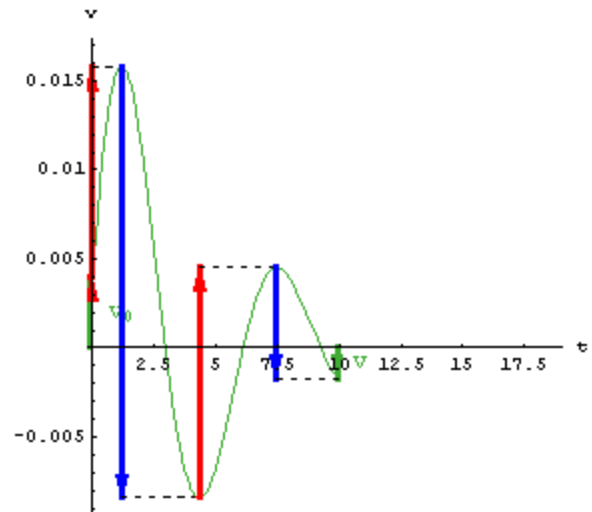
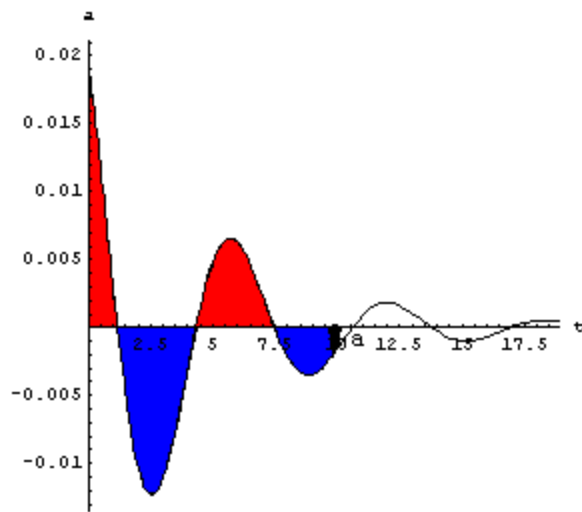


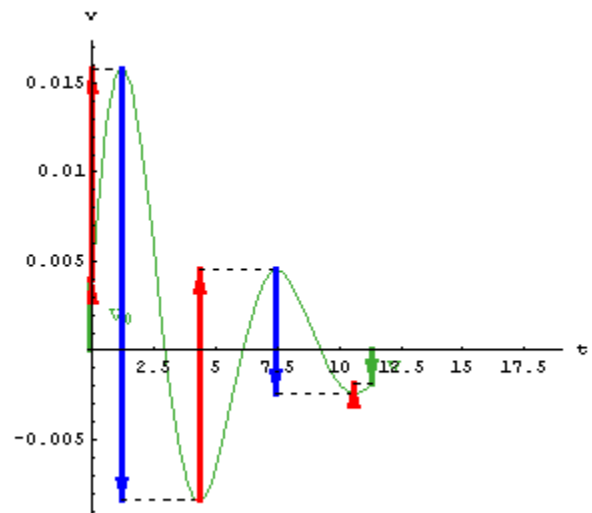
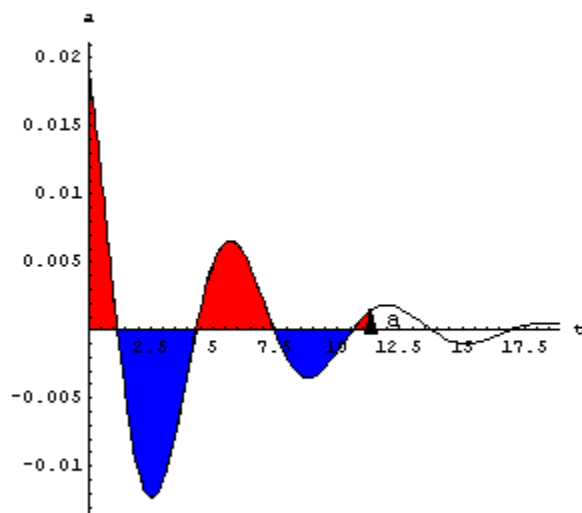
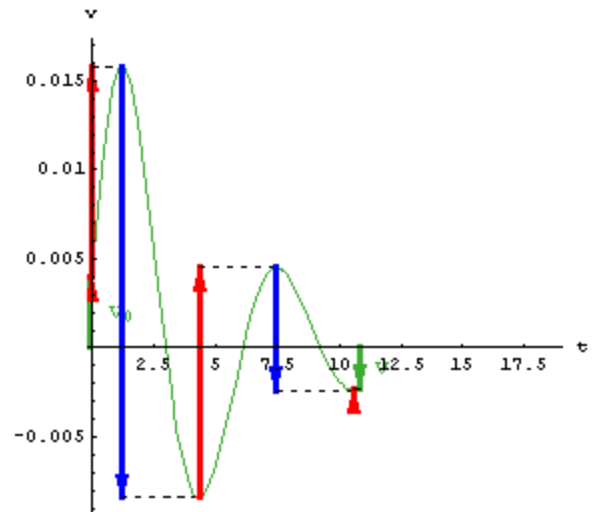
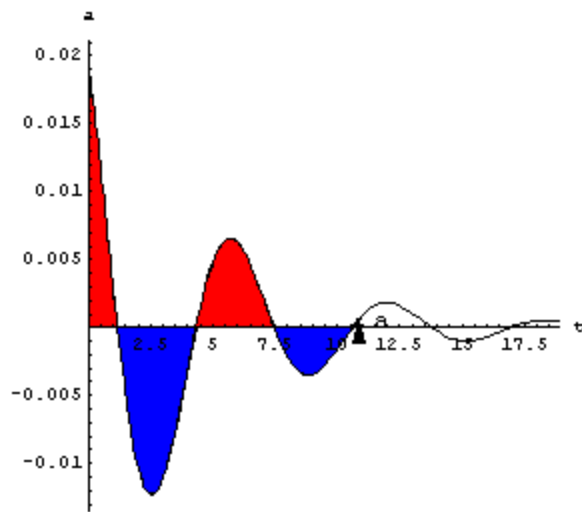


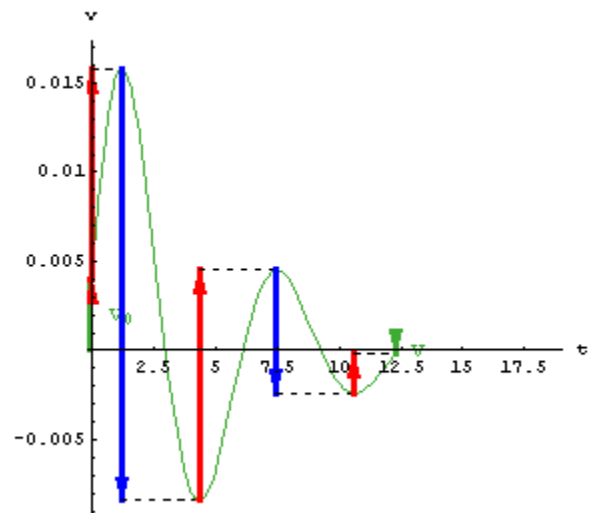
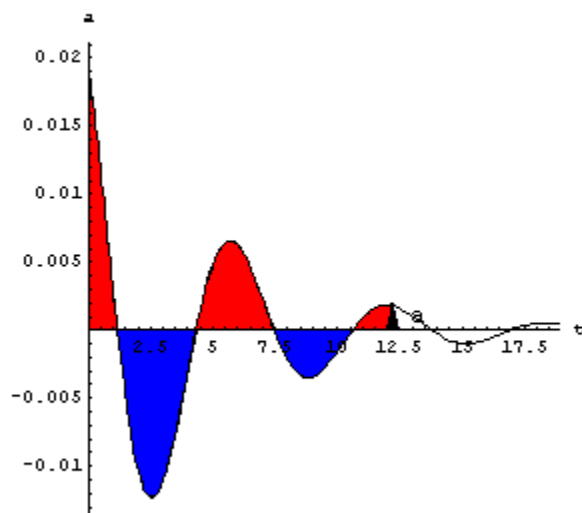
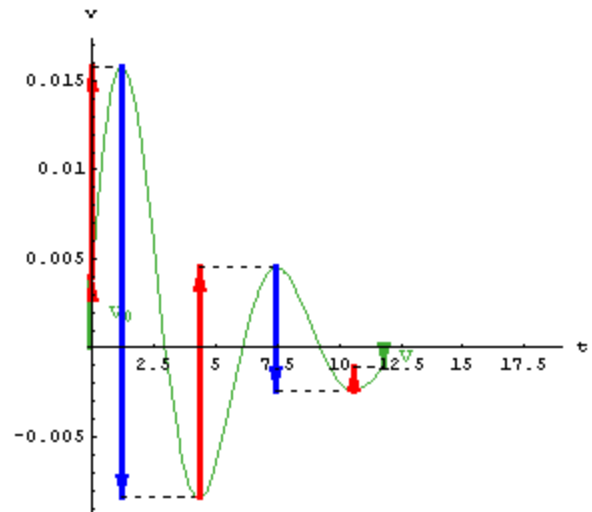
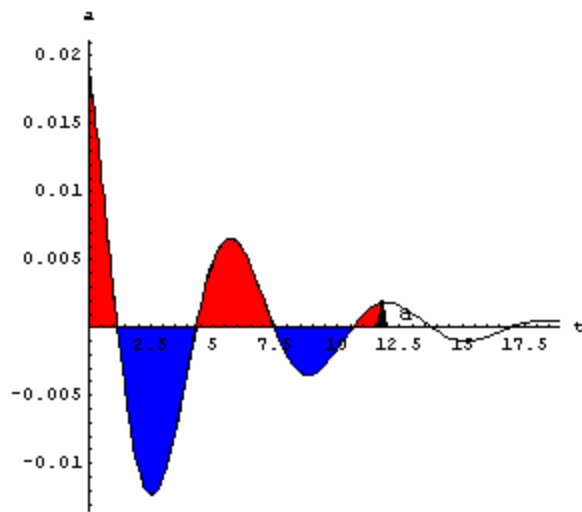


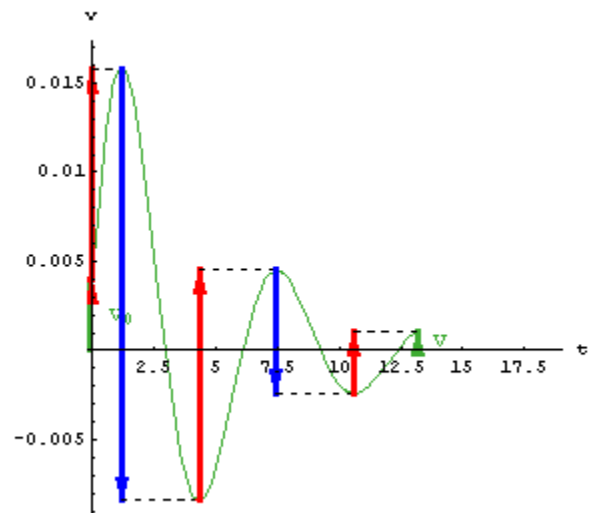
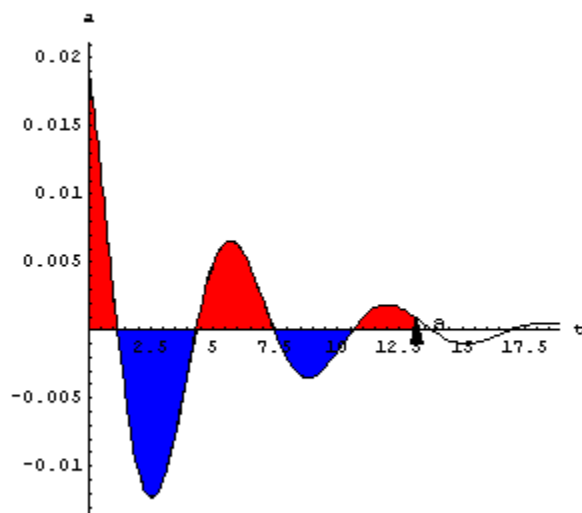
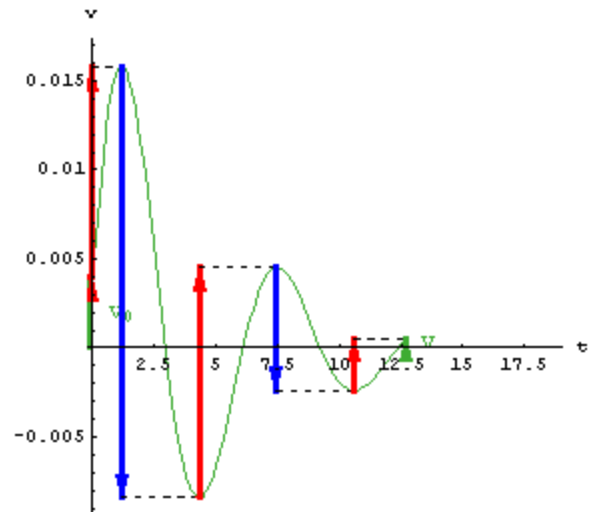
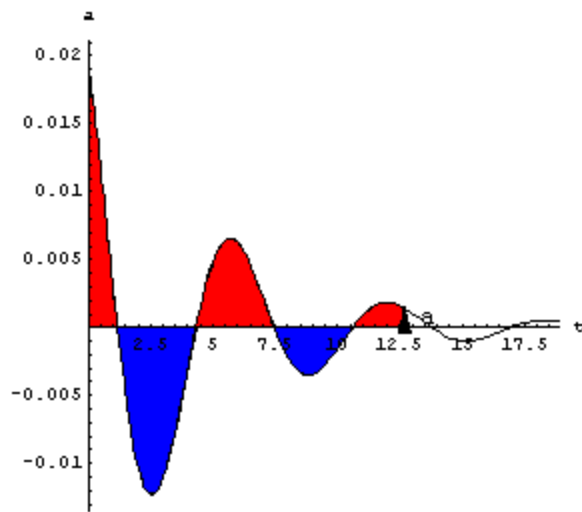


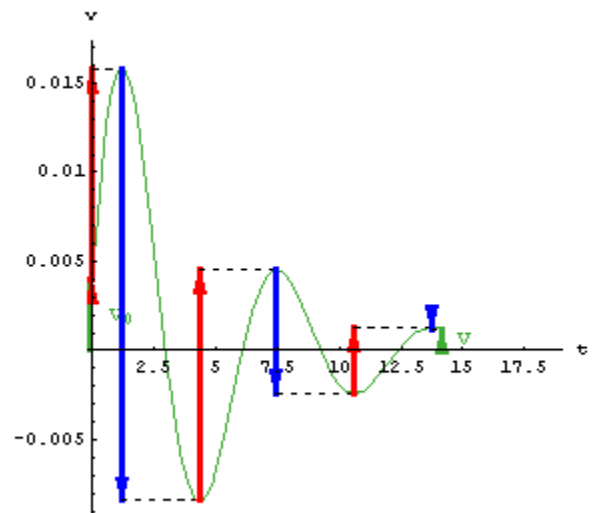
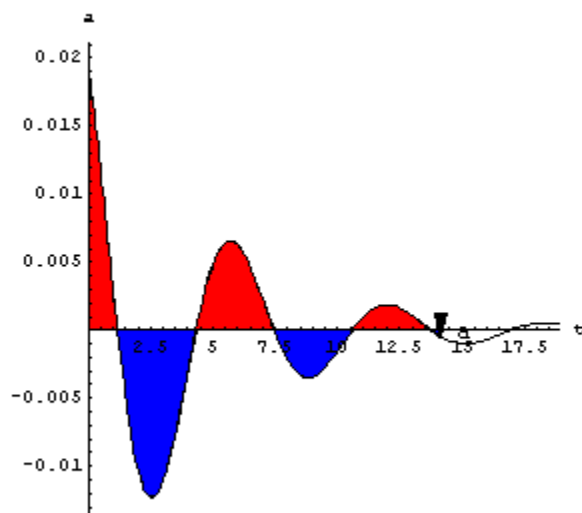
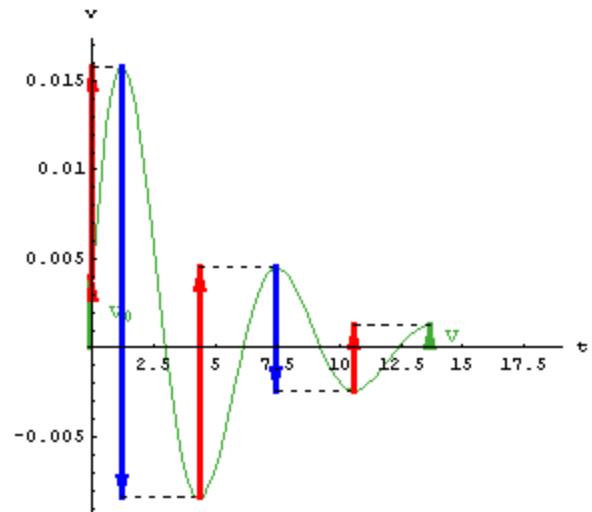
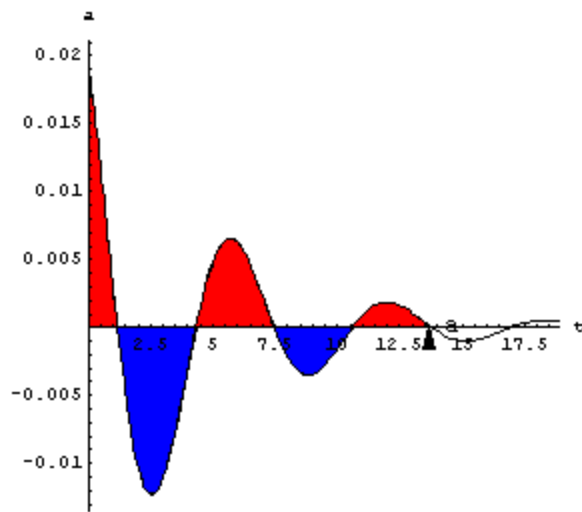


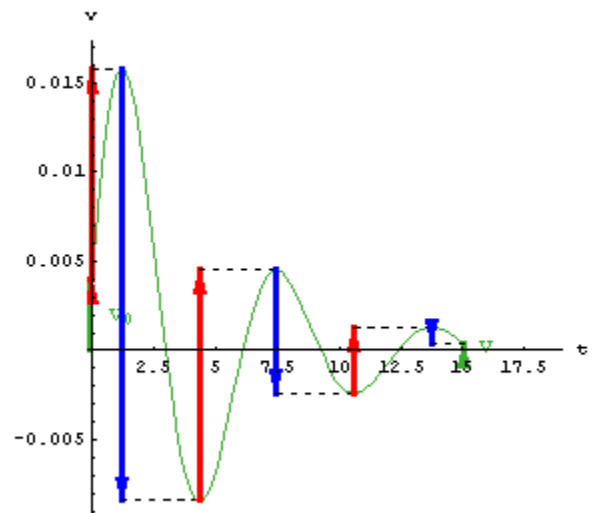
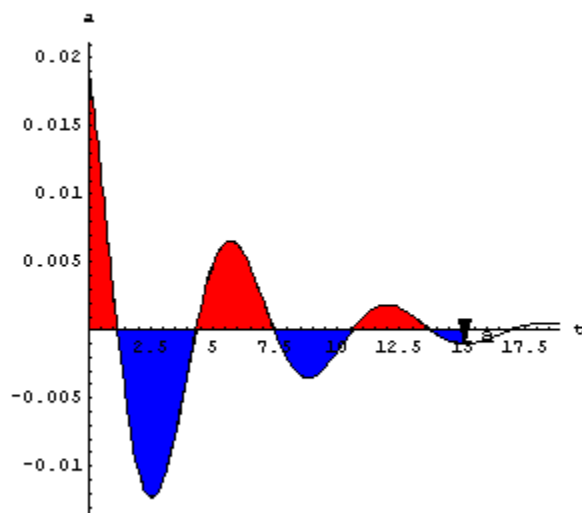
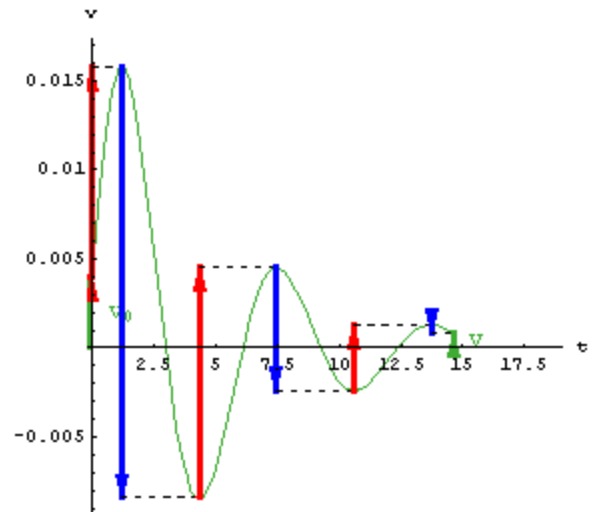
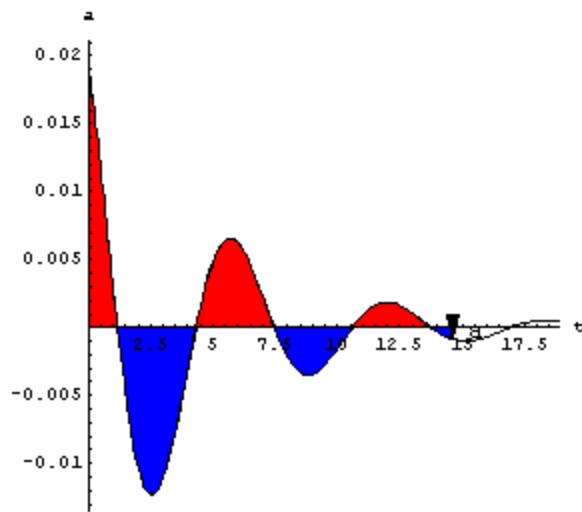


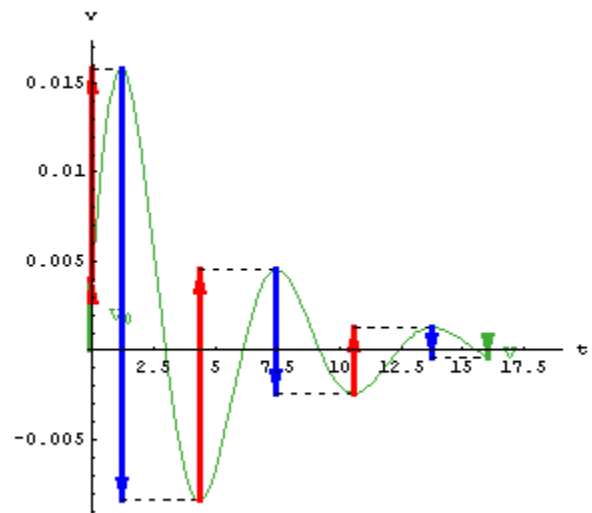
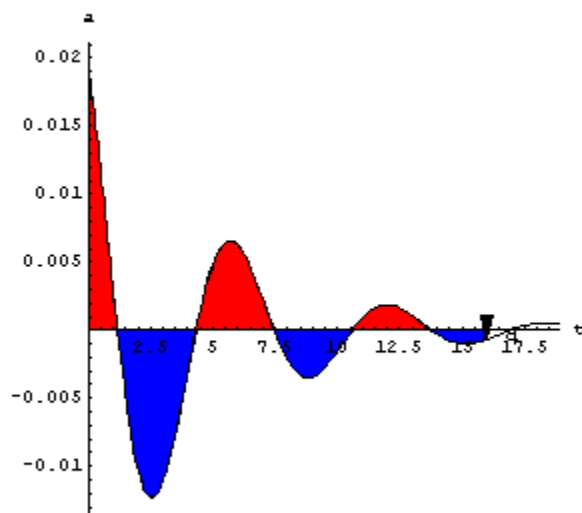
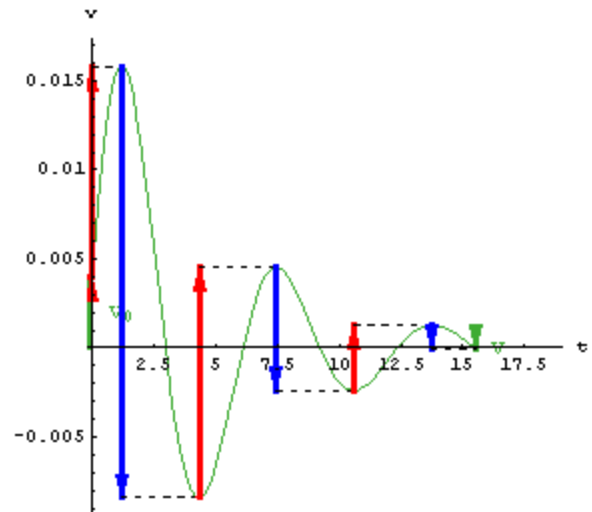
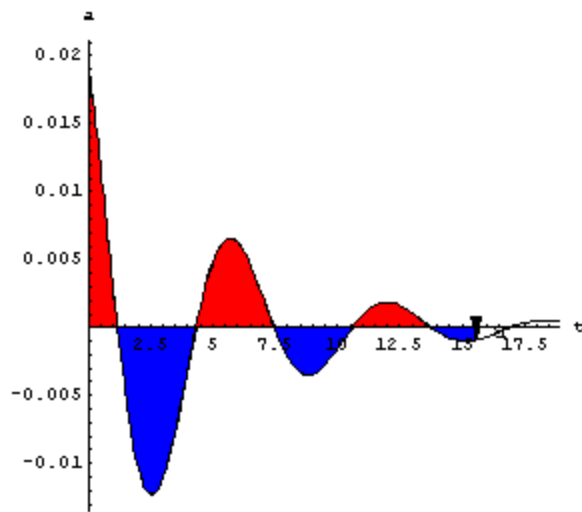


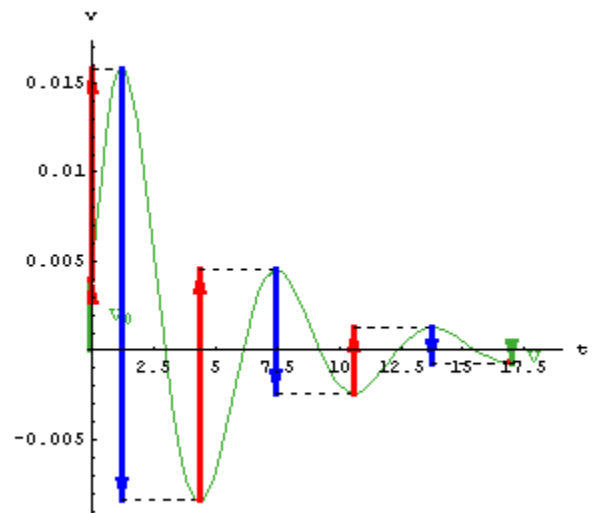
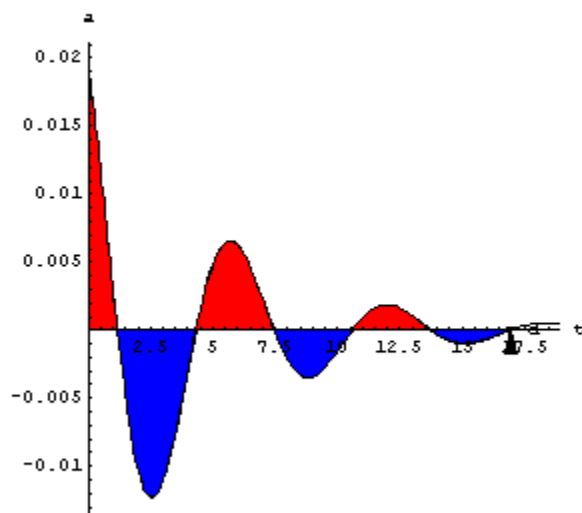
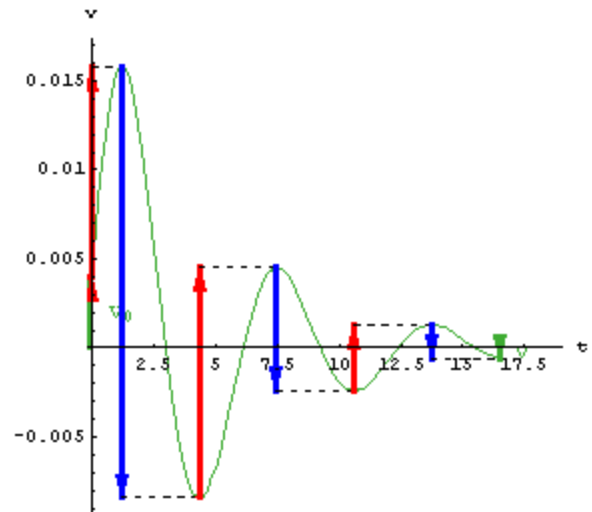
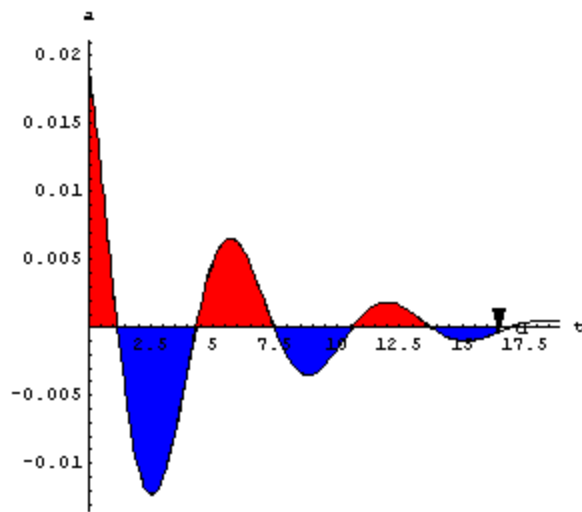


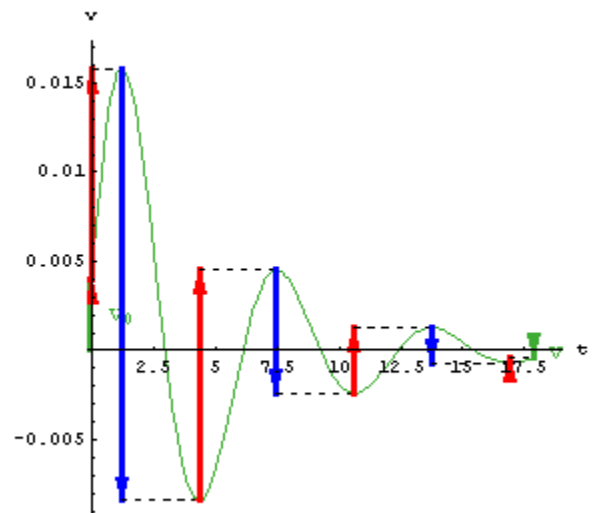
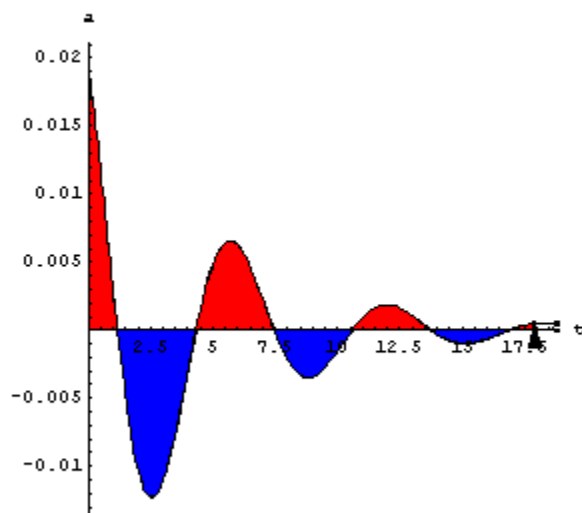
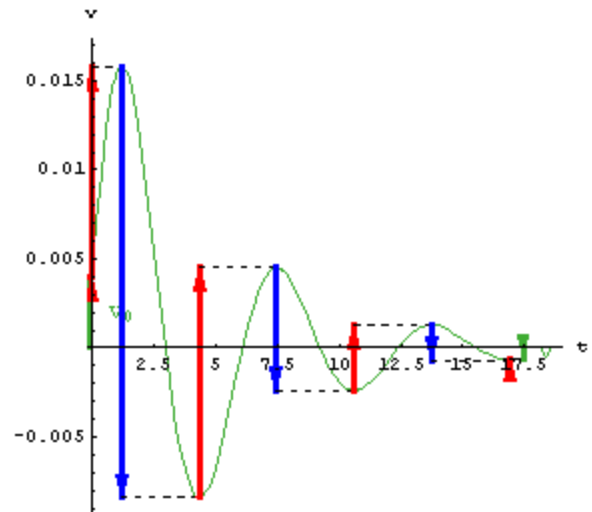
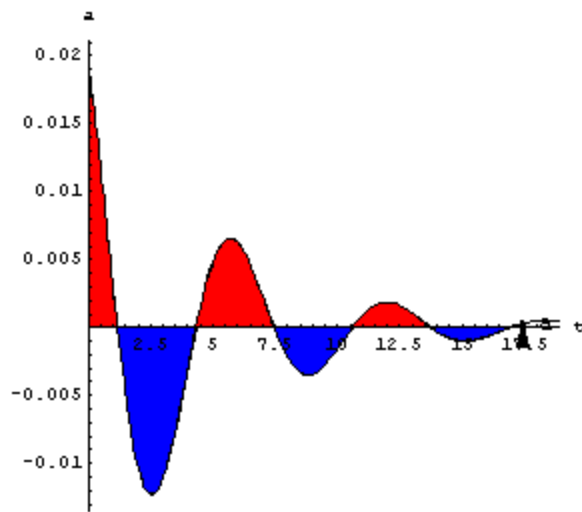


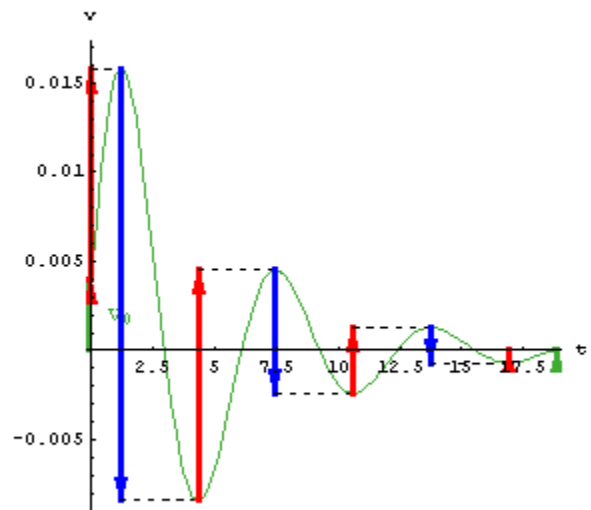
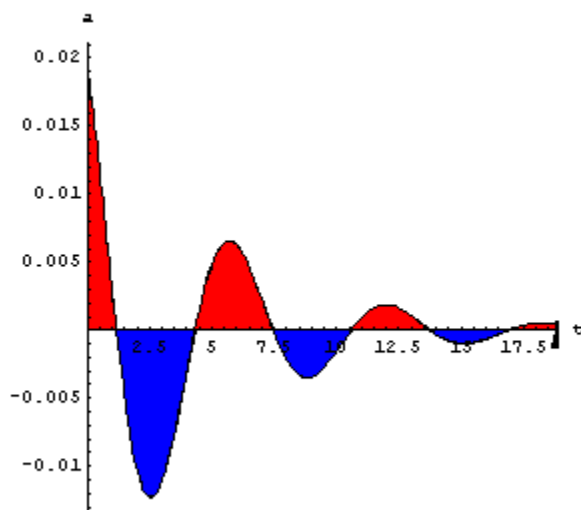
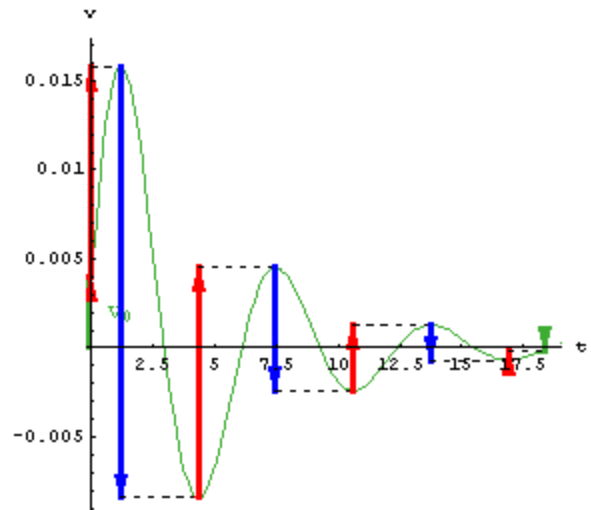
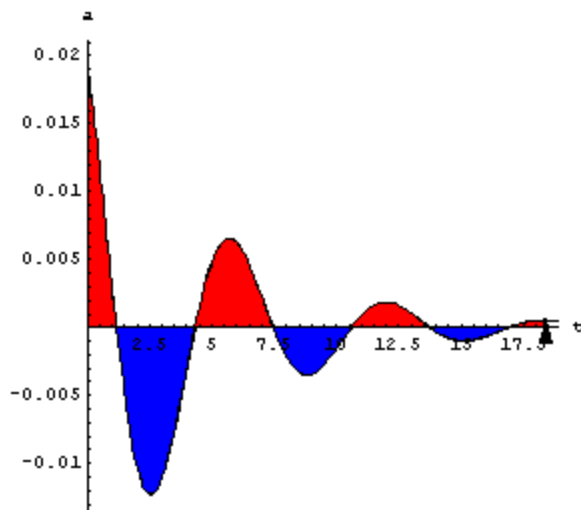












□ About *Mathematica*

The **Which[]** command is useful for creating piecewise defined functions. To learn more about the **Which[]** command, pull down the Help menu, select the Help Browser, and type **Which**.

[Go Back.](#)

Mathematica graphs a function by plotting a series of points on the function and then

connecting them with straight lines, sometimes when they really shouldn't be connected. The vertical line at $t=4$ hours really shouldn't be there. The upper segment of the graph at $+60$ mph should not be connected to the lower segment at -60 mph. [Go Back.](#)

Created by [Mathematica](#) (August 9, 2005)

