The harmonic oscillator with energy conservation

If we follow the motion of a harmonic oscillator for a long time (~1000 oscillations), the amplitude may decay due to numerical errors. This decay (observable in the first Harmonic Oscillator exercise) will be prevented in this exercise by incorporating conservation of energy into numerical model at the outset.

Consider a one-dimensional electrostatic potential well: $\phi(x) := \alpha \cdot x^2$

A particle with charge q will have a potential energy PE: $PE(x) := q \cdot \phi(x)$

The kinetic energy KE is defined using the velocity v and mass m: $KE := 0.5 \cdot m \cdot v^2$

The total energy W is a constant. We can find the kinetic energy at any location using conservation of energy: $KE := W - PE(x)$

We can solve for the velocity as a function of x: $v(x) := \sqrt{\frac{2}{m} \cdot (W - PE(x))}$

But it could also be with a minus sign: $v(x) := -\sqrt{\frac{2}{m} \cdot (W - PE(x))}$

The position of the particle can be found by integrating the velocity.
There will be a problem deciding which sign of the square root to use.

We will begin by assigning q = 1 and m = 1.
We will also let α = 0.5 and W = 0.5, then v becomes the function: $vfcn(x) := \sqrt{1 - x^2}$

The electric field for this system is -x, thus the "spring constant" (force/distance) is unity and the frequency of oscillation is unity.

Initial conditions: The particle will start at the origin with unity velocity.

$x0 := 0$ $v0 := 1$

Time step: The equation of motion will be iterated with a time step of Δt: $\Delta t := \frac{\pi}{15}$
One period of oscillation (2π) will require 30 time steps.

Finding the sign of the square root function:

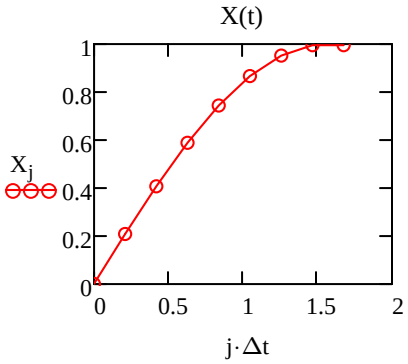
The harmonic oscillator problem has an analytic solution. We will make a plot of the solution to help us find a method of deciding the sign of the velocity.

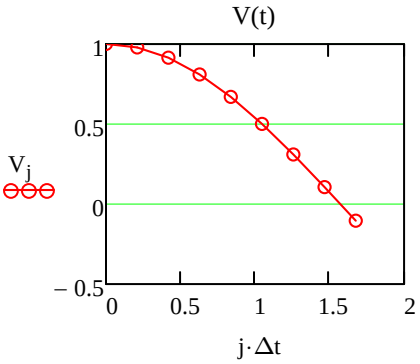
$j := 0 .. 8$ An index for values of X and V.

$X_j := \sin(j \cdot \Delta t)$ Analytic value for distance X.

$V_j := \cos(j \cdot \Delta t)$ Analytic value for velocity V.

Plots of the analytic solutions for X(t) and V(t):





Inspection of the plots shows that the X position lingers at the turning point, making it hard to decide whether or not the velocity has changed. This decision is even more of a problem if we are not sure of the x value for the turn around. The velocity V(t) behaves almost linearly when it passes through zero. This behavior suggests that we could extrapolate V(t) linearly, and use the extrapolated value to decide the sign of V. If our extrapolated velocity is very near the turning point V = 0, then we could be wrong about the sign if the true value is on the other side of the zero from the extrapolated value. However, in this case the velocity is very near zero, so if we get the sign wrong, the magnitude of the error is small. For example, if we extrapolate to 0.001 and the true value is -0.001, the error is only 0.002.

Extrapolation of V(t):

We will extrapolate to the next value by writing:
We have switched to lower case v to avoid confusion with V.

$$v_j := v_{j-1} + \Delta v$$

We will find Δv from the previous values for v, using:

$$\Delta v := v_{j-1} - v_{j-2}$$

Combining these we get the extrapolation:

$$vextrap := 2 \cdot v_{j-1} - v_j$$

Starting values:

If we are going to use linear extrapolation, we need to have two consecutive starting values for x and v:

$x0 := 0$ $\quad$ $v0 := vfcn(x0)$

$x1 := x0 + v0 \cdot \Delta t$ $\quad$ $v1 := vfcn(x1)$

These starting values will be calculated again in the program loop.

Successive approximations:

Once we have found the trial value for the next x_j, we can use the trial x_j to find the next v_j as vfcn(x_j). The "best" value for velocity between the points x_j and x_{j-1} will be the average of the values for v at these two points because this gives us "centered" finite differences. Hence the velocity for the extrapolation will be 0.5 (v_j+v_{j-1}).

Avoiding a complex root:

If our extrapolated value for x is slightly greater than unity, the function v(x) will be a complex number. To avoid this, we will redefine the function v(x) so that it returns zero for x>1. This change will not do much harm because v is near zero if x is near unity.

$$vfcn(x) := \text{if}\left(1 - x^2 < 0, 0, \sqrt{1 - x^2}\right)$$

Program loop:

A program loop will find the values for x_j and v_j by the following procedure:

1) Linearly extrapolate v to find vextrap, the trial value for v_j.

2) Use a **parabolic** extrapolation to find the next value of x as

xextrap = x_{j-1} + vextrap*Δt + 0.5*a*Δt^2. The acceleration a = (v_{j-1} - v_{j-2})/Δt.

Linear extrapolation is not accurate at the turning point because a is large and v is small.

3) Use vfcn(xextrap) to calculate the new v_j, and use the sign of vextrap to assign the sign.

4) Use the new v_j to calculate x_j again by parabolic extrapolation.

5) Place the x values in an array A.

Time step: $\Delta t = 0.209$ This time step is about 0.2 radian and there are 30 steps in a period.

$jmax := \frac{20 \cdot \pi}{\Delta t}$ $jmax = 300$ This number of time steps will result in 10 oscillation periods.

$A :=$

$x_{jmax} \leftarrow 0$

$v_{jmax} \leftarrow 0$ Initialize the x and v vectors.

$x_0 \leftarrow 0$

$v_0 \leftarrow vfcn(x_0)$ Assign the starting values.

$x_1 \leftarrow v_0 \cdot \Delta t$

$v_1 \leftarrow vfcn(x_1)$

for $j \in 2 .. jmax$

$\quad vextrap \leftarrow 2 \cdot v_{j-1} - v_{j-2}$

$\quad xextrap \leftarrow x_{j-1} + v_{j-1} \cdot \Delta t + 0.5 \cdot (vextrap - v_{j-1}) \cdot \Delta t$ This is the program loop described above.

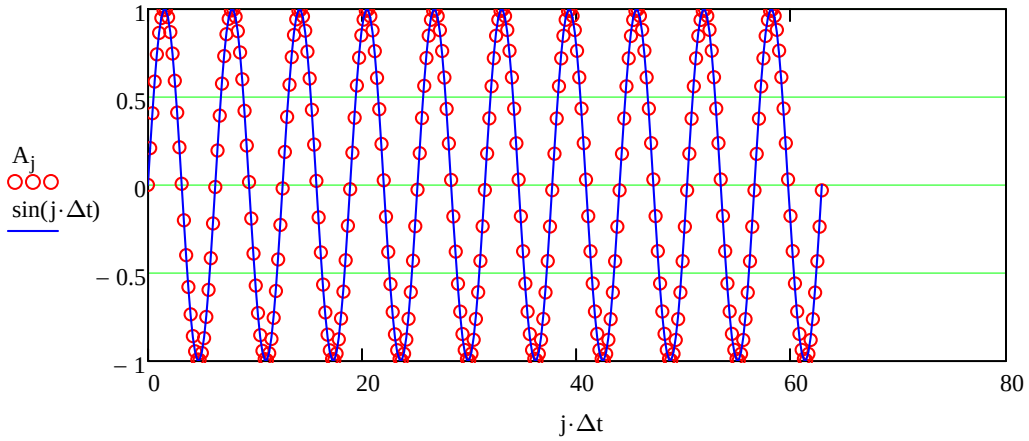
$\quad v_j \leftarrow sign(vextrap) \cdot vfcn(xextrap)$

$\quad x_j \leftarrow x_{j-1} + v_{j-1} \cdot \Delta t + 0.5 \cdot (v_j - v_{j-1}) \cdot \Delta t$

x

Plot the computed (circles) and analytic (line) solutions:

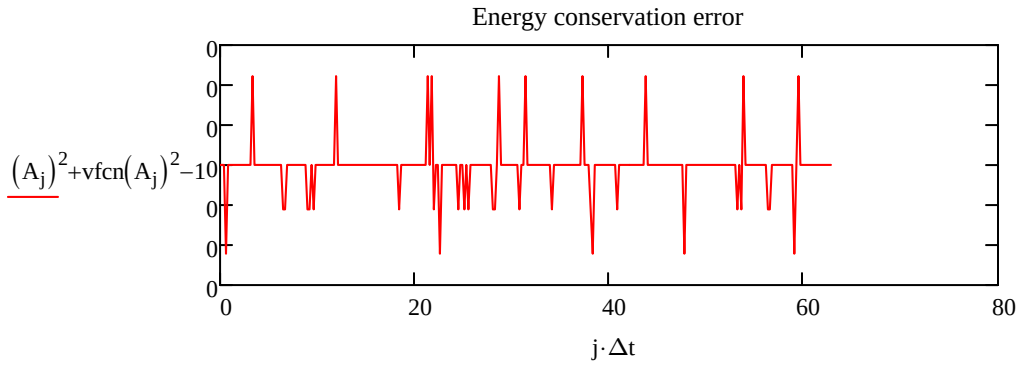
$j := 0 .. jmax$ is the counting index.



Note that there is a tiny error in the phase after 10 oscillations. We expect the final point to be at x = 0 if there are exactly 10 periods. Instead we find:

$A_{jmax} = -0.031$ which indicates a small phase error.

Conservation of energy should guarantee that we have no amplitude error. Energy conservation for this problem corresponds to $x^2 + v^2 - 1 = 0$. We plot this sum below:



This shows that energy conservation is exact except for roundoff error, and this error does not accumulate with time. This plot will look different for other values of time step.

Try it: Plot the accumulated phase error as a function of time.
(For example, plot the 30th, 60th, 90th time steps, etc.) Does this error increase in time? Does this error diminish if the time step is reduced?

Try it: What happens if the time step is $\pi/14$? $\pi/17$ $\pi/5$? $\pi/4$? Is there more or less phase error if some $j*\Delta t$ values lie precisely at maxima?