Trebuchet Mechanics

Donald B. Siano

dimona@home.com

Mar 28, 2001

Abstract

The trebuchet, a medieval catapult driven by a falling, hinged counterweight, has been simulated to progressively more accurate approximations by successively more realistic physical models. The first of these, a "black box" model in which the mechanism for the transfer of the potential energy of the counter-weight to the kinetic energy of the released projectile is left unspecified, led to a definition of a "Range Efficiency," Reff, equal to the measured range of the projectile divided by the range of the black box model, given by twice the ratio of the CW to projectile masses times the distance that the CW falls. This range efficiency can be used to compare actual trebuchets to simulated ones, design "back of the envelope" trebuchets, and is useful in understanding the results of more sophisticated simulations.

The method used to arrive at the most accurate simulations is to use the Mathematica programming language to derive the three coupled differential equations from the Lagrangian for the system. This method allows the sliding constraint equations to be readily derived and translated into a form useful for other, faster, computer languages. The equations, while consisting of many terms, are amenable to solution by Mathematica's differential equation solver, as well as by a fourth-order Runge-Kutta method. The solution has been verified by various methods, including one that involves using the solution to calculate the total energy of the system at every instant and showing that it is a constant.

The physics of the release mechanism is also described in some detail, and the dependence of the range on the finger angle and the coefficients of friction is given in approximate terms. The benefits of propping the counterweight, as opposed to letting it hang freely from the end of the beam is also discussed.

A major result is the design of an efficient trebuchet, by exploring the design space by doing thousands of simulations. The result indicates that one in which the beam is initially at a 45° angle, the sling is equal in length to the length of the long arm of the beam, and the long arm of the beam is four times as long as the short arm, is a reasonably efficient one, and is therefore recommended as a "nominal" design.

Introduction

The trebuchet is a medieval weapon of war--a catapult that is powered by a falling massive counterweight. Recently, it has undergone something of a revival of interest among historians, hobbyists, and assorted show-offs. While many have been successfully built with a rather wide variety of designs, most work on their design has been highly empirical--little work on the mathematical analysis on their operation and design has appeared. The object of this work is to obtain a fairly complete analysis of the device, so that the ingenuity of the medieval engineers can be more fully appreciated, and modern dabblers in the art can produce more reliable and powerful designs.

We will first describe the geometry of the full model, including the hinged counterweight and the sliding sling. Since the physics is rather complicated, however, our approach will be to first briefly describe several simplified versions of the machine, with each successive model more closely approximating the real device.

The Geometry

Refer to the diagram shown in Fig. 1 to see the definition of the parts of the trebuchet and the angles used to define the configuration. It is shown here in an assumed initial configuration, with the origin taken at the pivot of the beam. The counterweight is hinged and has a center of mass at a distance l4 from the end of the short arm of the beam. The beam is of uniform cross-section and has a mass mb. The mass of the projectile is m2 and it is at the end of a weightless sling at a distance l3 from the end of the long arm of the beam, which has a length l2, as shown. The three angles required to describe the motion are (θ, φ, and ψ). The main object of the simulation is to calculate the values of these angles and their derivatives as a function of time from the initial values of the angles and the values of the eight parameters (l1, l2, l3, l4, l5, m1, m2, mb).

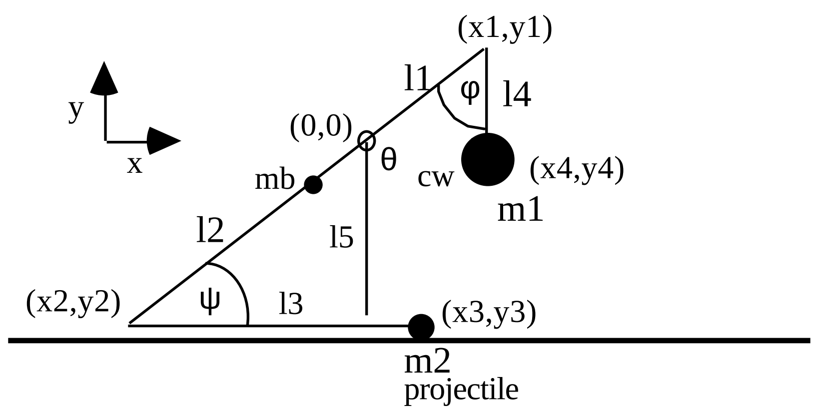


Fig. 1 The geometry of the trebuchet, showing the three angles taken as the independent variables, in a configuration at the start of the movement.

Operation of the Trebuchet

Upon release of the trigger holding the beam in its starting position (usually placed near the end of l2), the counterweight falls, and in the first part of the motion, the projectile in the sling slides along a horizontal trough beneath the beam. As the counterweight accelerates, the projectile leaves the trough, swinging freely through an arc. At some point a mechanism of some sort (usually a peg and hook arrangement) releases one end of the sling, freeing the projectile which then flies freely to the target.

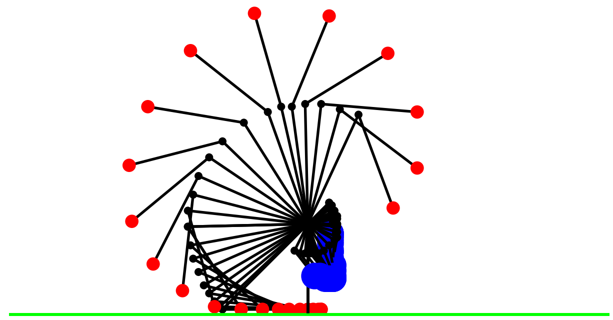


Fig. 2. The motion of the trebuchet at constant time intervals.

Limitations of the model:

All of the models to be described differ from real devices in two important respects. First, all of the parts are assumed to be rigid and the joints rotate perfectly around points. The model is assumed to be rigidly fastened to the ground. In reality, there will be some flexibility in all of the parts, and often, the model just sits on the ground. Second, we assume all of the parts are without friction. The projectile during the throw experiences air resistance, and there is some unavoidable friction at axle.

While some of these effects (friction at the axle and the air resistance during the flight of the projectile) could be readily added to the model, for many purposes we would prefer a simpler model, with fewer adjustable parameters. The model is more a tool for gaining a qualitative understanding, than one for precisely replicating the results obtained on some particular embodiment.

Some Elementary Analysis of the Trebuchet

Consider a projectile fired on a horizontal plane that has a velocity v_o at an angle α with respect to the horizontal. It will have a range given by

$$R = \frac{2\, v_0^2 \sin\alpha \, \cos\alpha}{g} ,$$

where g is the acceleration due to gravity. This has a maximum range for $\alpha = 45°$ which is

$$R_m = \frac{v_0^2}{g} .$$

The kinetic energy in the projectile, having a mass m_2, at the start of the trajectory is then

$$KE_{proj} = \frac{m_2 v_0^2}{2}.$$

A counterweight mass m_1 at a height h above a reference plane has a potential energy given by

$$PE_{cw} = m_1 g h.$$

The most efficient mechanism for a trebuchet would, clearly, be able to transform all of the initial potential energy into kinetic energy of the projectile. Assume that there is a perfectly efficient mechanism that can do this--a sort of "black box". The geometry is shown in the Fig. 3.

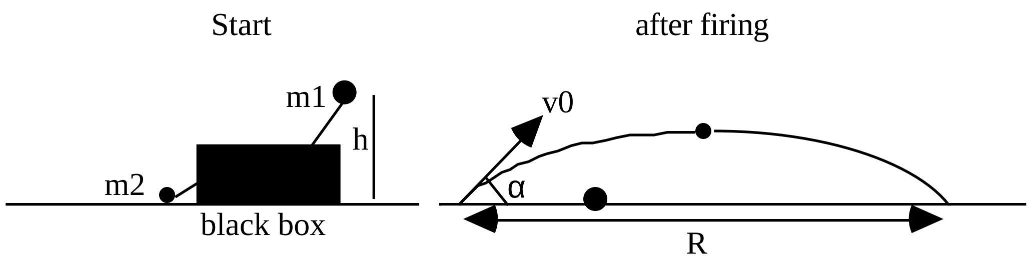


Fig. 3. The black box trebuchet with with the range and intial conditions shown

Thus, if the mass of the counterweight is initially at a height h above its lowest point, the maximum possible range that could be attained is obtained by equating the initial potential energy in the counterweight with the kinetic energy in the projectile at the start of the trajectory, yielding

$$R_m = 2 \frac{m_1}{m_2} h .$$

This theoretical maximum range is easily seen to be reasonable--it is larger for heavier counterweights and lighter projectiles. It is linear in the initial height of the counterweight. Perhaps surprisingly, depending upon one's mechanical intuitions, it is independent of the acceleration of gravity--it would throw just as far on the moon!

This simple equation can be very useful in the preliminary design of any treb. Decide on how high you are willing to lift the counterweight, and you have an estimate of how far you can throw a projectile of a certain mass....

A real trebuchet will not, of course, attain this theoretical maximum range because of various factors such as the friction at the axle, the slide, the air resistance on the projectile, rotational energy of the beam, and unutilized kinetic energy remaining in the swinging CW after the projectile is fired.

Range and Energy Efficiency

The efficiency of a real trebuchet can be reasonably defined in at least two ways. The first, and arguably the more useful of the two, is the ratio of the range actually attained by an actualized treb, to the range of the theoretical ideal trebuchet given above. This is termed the "range efficiency", ε_R. The range efficiency of a model with no air resistance will be given by

$$\varepsilon_R = \frac{R}{2\frac{m_1}{m_2}h} = \frac{m_2}{m_1}\frac{v_0^2 \sin\alpha\cos\alpha}{g\,h},$$

where h is the distance the counterweight can drop, and α is the angle that the projectile makes with the horizontal at the start of its flight after leaving the trebuchet.

The distance that the counterweight can fall, given by a little geometry, is readily seen to be

$$h = l_1(1 - \cos(\theta_z)) = l_1(1 + \sin(\psi_z))$$

when the sling is initially horizontal and the counterweight hangs vertically. It is independent of l2 and, perhaps surprisingly, l4 .

When the CW is initially propped, the initial angle between the CW and the beam is needed in order to calculate h. In this case, the distance that the CW can fall is given by

$$h = l_1(1 - \cos(\theta_z)) + l_4(1+\cos(\theta_z + \varphi_z))$$

where θ_z and φ_z are the initial values for θ and φ, respectively. There is some interest in propped counterweights, and their impact on efficiency, range and the shock experienced by the trebuchet when the CW is dropped, tending to tear the machine apart. We will discuss some of this in a later section.

The range efficiency of a real treb is easy to measure (measure R, m_1, m_2 and h) and calculate for particular models. It will, of course, be less than one. For trebuchets that misfire ($\alpha > 90°$), it could be less than zero!

Another possible useful definition of an efficiency for a trebuchet would be the fraction of the potential energy in the counterweight that is <u>actually</u> deposited as kinetic energy in the projectile. Call this the energy efficiency, ε_E.

It is given by

$$\varepsilon_E = \frac{E_{ke}}{E_{pot}} = \frac{m_2\,v_0^2/2}{m_1\,g\,h}.$$

The energy efficiency is always greater than zero, and is therefore not equivalent to the range efficiency given above. The relationship between the two measures is readily seen to be

$$\frac{\varepsilon_R}{\varepsilon_E} = 2 \sin \alpha \cos \alpha .$$

Thus, when $\alpha = 45°$,

$$\frac{\varepsilon_R}{\varepsilon_E} = 2 \sin 45° \cos 45° = 1$$

and the two efficiencies are equal. The optimum release point for the projectile to achieve a maximum range is seldom exactly 45°, so there is usually a small difference between the two efficiencies--the range efficiency is generally slightly smaller than the energy efficiency.

There is actually at least one other possible viewpoint on efficiency which would include the beam as a part of the driving energy that should be considered. Our method, discussed above, views only the CW as the "fuel"--the potential energy stored in the cocked beam is left out. The range efficiency definition is not only conceptually simpler, it is much easier to compute in practice. Very few builders know the position of the center of mass of the beam, its mass, or its radius of gyration. And since the energy of the CW is usually much larger than that in the beam, the difference in the efficiency is not very large either. Better to keep it simple.

The major factors contributing to inefficiency of a real trebuchet are that due to the kinetic energy in the beam and the counterweight at the time for optimum release of the projectile. If the projectile could be released when the counterweight is at its lowest point and has a very low velocity, then the efficiency would be relatively high. This is the function of the sling and the CW hinge, and as we shall see, the efficiency of real trebuchets can be amazingly (for a medieval engine) high.

Dimensional Analysis

It is easy to see by a conventional dimensional analysis of the model (as described in the figure) that the solutions of the equations can be put into the form

$$R = l1\ f(l2/l1,\ l3/l1,\ l4/l1,\ m2/m1,\ mb/m1, \theta_S, \varphi_S, \psi_S)$$

where f is some unknown function, and the s subscripts refer to the values of the angles at the start. Note that the range is invariant to the acceleration due to gravity, g. More importantly, this result allows us to greatly reduce the number of solutions to the equations that need to be obtained to see representative behavior. If all of the lengths (l1, l2, and l3) and masses are doubled, for example, while the starting angles are unchanged, the range will also be doubled--the function f is unchanged by this transformation, but l1 is doubled.

A similar approach for the times during the motion for different models will clearly go as

$$T = \text{Sqrt}[ll/g]\ F(l2/l1,\ l3/l1, l4/l1, m2/m1,\ mb/m1, \theta s,\ \varphi s,\ \psi s),$$

where F is some unknown function. The time from the beginning to the release thus depends on g--a similar model with the masses and lengths all doubled, while the starting angles are unchanged, would have the times increased by the square root of two.

While useful, this is not the whole story on scaling up models. It is important to recognize that larger trebuchets have larger counterweights--the counterweight box containing the stone increases in volume as the cube of the dimensions, so the mass of the counterweight and the lengths do not increase proportionately.

Now consider a treb with dimensions l and another dimensionally similar one with dimensions L. That is, the height at the axle, height of the cw at the start, beam length and sling length all scale as L/l. The angles between the parts of each trebuchet are the same in both. In real trebuchets, the weight of the counterweight and projectile would (assuming they are made out of the same stuff, having the same density) scale as $(L/l)^3$. The potential energy in the cw at the start is proportional to the height and the weight of the counterweight, so it scales as $(L/l)^4$. The force required to cock the trebuchet will scale as $(L/l)^2$ because the weights scale as the cube, but the lever arms scale as L/l, reducing the force required.

As shown above for the "black-box" model, and the dimensional analysis result, the range will clearly scale as (L/l), because h is proportional to l. Thus, doubling the size of a treb allows one to throw a projectile 8 times as heavy twice as far. The larger trebuchet would contain 16 times as much energy as the smaller one. This is why trebuchets are big.

The see-saw trebuchet

It is instructive to analyze a slightly more realistic version of the trebuchet--one that eliminates the sling, but has a counterweight attached at one end, as shown in the figure.

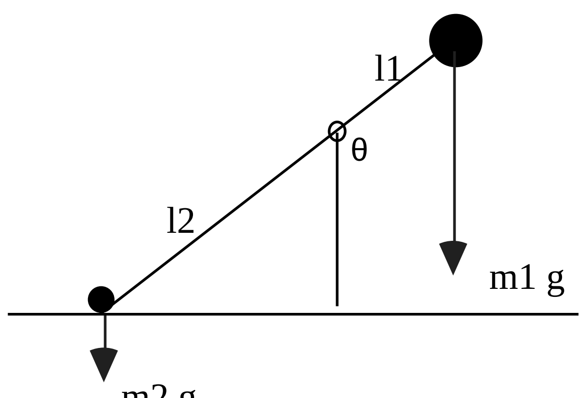


Fig. 4. The "see-saw' trebuchet. It has a fixed counterweight and no sling.

It has a massless beam as well. During the movement of the trebuchet, it will experience a torque due to the unbalanced weights, which has a magnitude

$$\tau = m_1\, g\, l_1 \sin\theta - m_2\, g\, l_2 \sin\theta$$
$$= g \sin\theta\, (m_1 l_1 - m_2 l_2)$$

The moment of inertia about the axle is simply

$$I = m_1 l_1^2 + m_2 l_2^2,$$

so the equation of motion is

$$I \ddot{\theta} = \tau$$

$$\ddot{\theta} = - \frac{g\,(m_1 - m_2)}{m_1^2 - m_2^2} \sin(\theta) = c \sin(\theta)$$

where c is a constant. A negative sign was introduced in the latter equation because the torque was defined to be positive in the clockwise direction, the direction of decreasing θ. Note that when θ is zero, the trebuchet is pointed straight up, and the behavior beyond this point is of little interest.

As an example, to be followed closely in the successive models, we will take some typical values one might use for a model design: m1=100 lb, m2= 1 lb, l1= 1 ft, l2 = 4 ft, g = 32 ft s^{-2}. Then c = 26.48. Use the initial condition that the start of the movement the beam is stationary and θ is 135°. We suppose for now that the projectile releases at θ = 45°. The velocity of the projectile is then

$$v_0 = l_2 \dot{\theta}$$

evaluated at the time when $\theta = \pi/4$.

This can be readily solved to get θ and its derivative as a function of time by a variety of numerical methods and programming languages. The Mathematica programming language from Wolfram Research Inc. is a particularly interesting route to take because it has an unusual combination of strengths in mathematical symbol manipulation (it will be used to derive the equations for the more complex models) and powerful numerical routines for solving differential equations.

This example is solved numerically in the Mathematica programming language and the program with the output is shown in Appendix 1. We see that the range is 37.5 ft when the release is at 45.0°, and so the efficiency is 11.0%. The maximum range is obtained when the release is a little later than this: when θ is 38.0° it is 38.4 ft. The increase in the velocity due to the acceleration of the counterweight overcomes the less-than-optimal release angle of 45°. We can do much better with the addition of the sling and a hinged counterweight.

The See-Saw with a Hinged Counterweight and No Sling

The preceding example can be made a little more realistic by arranging for the counterweight to be hinged, rather than fixed to one end. We suppose for our example that it is attached by a rod of length l4.

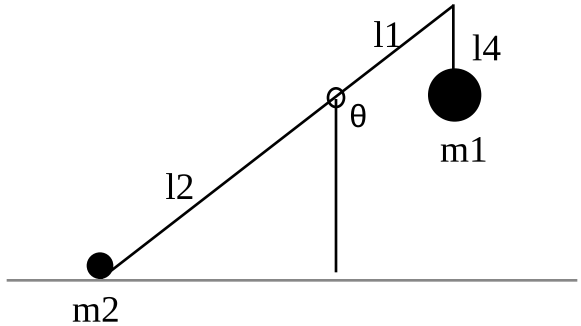


Fig. 5. The see-saw with a hinged counterweight.

This model requires a different approach to get the equations of motion--analysis by torques and forces gets quite awkward. An easy path is to use the method of Lagrange. In this approach, one requires the kinetic and potential energies of the system as a function of the coordinates. The kinetic energy for the system (with a uniform beam of mass mb) can be obtained by elementary methods to be

$$T = \frac{m_1}{2}\left((\dot{x}_4)^2 + (\dot{y}_4)^2\right) + \frac{m_2}{2}\left((\dot{x}_2)^2 + (\dot{y}_2)^2\right) + \frac{m_b}{6}(l_1{}^2 - l_1 l_2 + l_2{}^2)\,\dot{\theta}^2$$

and the potential energy is

$$V = m_1\, g\, y_4(\theta,\varphi) + m_2\, g\, y_2(\theta) - \frac{m_b g\,(l_1 - l_2)}{2}\cos(\theta).$$

We choose to use θ and φ to be the coordinates that specify the configuration of the system, which are related to the Cartesian coordinates of the system as follows:

The Cartesian coordinates of the end of the short arm of the beam are

$$x_1(\theta) = l_1 \sin(\theta), \text{ and } y_1(\theta) = -l_1 \cos(\theta),$$

and, the coordinates of long end of beam, where the projectile is, are given by

$$x_2(\theta) = -l_2 \sin(\theta), \text{ and } y_2(\theta) = l_2 \cos(\theta).$$

The coordinates of the center of mass of the counterweight are

$$x_4(\theta,\varphi) = l_1 \sin(\theta) - l_4 \sin(\varphi+\theta) \text{ and } y_4(\theta,\varphi) = -l_1 \cos(\theta) + l_4 \cos(\varphi+\theta).$$

The Lagrangian is defined to be

$$L = T - V,$$

and we get the two simultaneous equations of motion by using the formulae

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{dL}{d\theta} = 0,$$

and

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} - \frac{dL}{d\varphi} = 0.$$

Though it is not difficult to do by hand, these equations are worked out in Appendix 2 using Mathematica. The results, in "fortranese" are

```
-l1 l4 m1 (phi'' + 2 th'') Cos[phi] +
       l1 l4 m1 phi' (phi' + 2 th') Sin[phi] +
       1/6 (6 l4^2 m1 phi'' + 6 l1^2 m1 th'' + 6 l4^2 m1 th'' +
            6 l2^2 m2 th'' + 2 l1^2 mb th'' - 2 l1 l2 mb th'' +
            2 l2^2 mb th'' + 6 g l1 m1 Sin[th] -
            6 g l2 m2 Sin[th] + 3 g l1 mb Sin[th] -
            3 g l2 mb Sin[th] - 6 g l4 m1 Sin[phi + th]) == 0,
```

and

```
   l4 m1 (l4 phi'' + l4 th'' - l1 th'' Cos[phi] - l1 th'^2 Sin[phi] -
         g Sin[phi + th]) == 0
```

Here, primes indicate derivatives with respect to time, and the angles (θ,φ,ψ) are, obviously, (th, phi, psi).

It is easy to see that when l4 = 0 and mb=0, they reduce to the single equation of motion given previously for the see-saw.

The values for the parameters (the lengths and masses) for the trebuchet can be plugged into these two equations, and they can be solved numerically for a suitable range of time. For our example we can use appropriate initial conditions: at t = 0, the values of θ and φ are taken to be $3\pi/4$ and $\pi/4$, respectively, so that the counter-weight hangs straight down; and the first time derivatives of θ and φ at t=0 are zero.

The solution for this model shown in Appendix II. It is interesting to compare the solution obtained by assuming that the release takes place at the time when $\theta = \pi/4$ with that for the see-saw model. The addition of the hinge to the counterweight increases the range, when the projectile is released at θ= 45°, from 37 to 70 ft, and so the efficiency roughly is doubled--a very significant improvement. A little reflection shows that this is mostly due to the decreased rotational kinetic energy in the counterweight allowed by the addition of the hinge.

The solution for the range as a function of the time of release is also shown in the Appendix II. One can see that the maximum range obtainable is much greater than that when the release is at $\theta = \pi/4$. The release can be delayed (in this example) to an angle of $\theta = 22.6°$. The time difference between these two angles is, however, only 0.024 s compared to the time for the throw of about 0.36 s. The range as a function of time is sharply peaked. To get the maximum range out of a trebuchet that follows this model would evidently require a very precise release mechanism .

The release mechanism can vary greatly from one trebuchet to another. Its behavior can be fairly complex, so for the time being we will just suppose for now that one exists that can be tuned to release it at any desired moment (configuration) in the throw. Our strategy will be to calculate the range for the projectile for every moment after the release of the trigger, plotting it, and taking the maximum range found as the quantity of primary interest.

The Treb with a Hinged CW and an Unconstrained Sling

The next most complex case to consider is to add a sling to the model just described. The projectile is now at (x3,y3) at a constant distance l3, from the end of the beam at (x2,x3). We can take l3 to be constant because the sling is under tension throughout the throw.

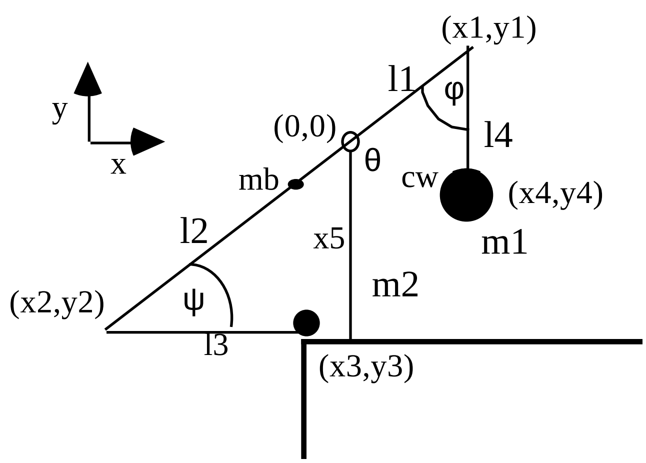


Fig. 6. The trebuchet with an unconstrained sling.

As before, the Cartesian coordinates of the parts of the treb with the angles and the various lengths can be readily derived, and are the same as above, except that the coordinates of the projectile are now given by

$$x_3(\theta,\psi) = -(l_3 \sin(\psi-\theta) + l_2 \sin(\theta)) \text{ and } y_3(\theta,\psi) = -(l_3 \cos(\psi-\theta) - l_2 \cos(\theta)),$$

where ψ is the angle between the beam and the sling.

We suppose for this model that the sling is initially horizontal under the beam, and falls freely into a slot, rather than sliding underneath the beam as in the usual actualization. This would be expected to be an approximation to the much more complicated case that involves the constrained sliding without friction.

The kinetic energy, T, and the potential energy, V, for this model of the trebuchet is the sum of the contributions from the counterweight, projectile and the beam:

$$T = \frac{m_1}{2}\left((\dot{x}_4)^2 + (\dot{y}_4)^2\right) + \frac{m_2}{2}\left((\dot{x}_3)^2 + (\dot{y}_3)^2\right) + \frac{m_b}{6}(l_1^2 - l_1 l_2 + l_2^2)\,\dot{\theta}^2$$

$$V = m_1\, g\, y_4(\theta,\varphi) + m_2\, g\, y_3(\theta,\psi) - \frac{m_b g\,(l_1 - l_2)}{2}\cos(\theta).$$

Since there are three degrees of freedom, the three equations for the (unconstrained) motion are

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{dL}{d\theta} = 0,$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} - \frac{dL}{d\varphi} = 0,$$

and

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\psi}} - \frac{dL}{d\psi} = 0.$$

These equations are now pretty lengthy. One can show (using Mathematica) that in "fortranese" the differential equation for θ is

```
     ((l1**2 - l1*l2 + l2**2)*mb*th'')/3. + (g*(l1 - l2)*mb*sin(th))/2. +
   -  g*m2*(-(l3*sin(psi - th)) - l2*sin(th)) -
   -  (m2*(2*(-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th))*
   -       (-(l3*(psi' - th')*sin(psi - th)) + l2*th'*sin(th)) +
   -      2*(-(l3*psi'*cos(psi - th)) +
   -         th'*(l3*cos(psi - th) - l2*cos(th)))*
   -       (l3*psi'*sin(psi - th) + th'*(-(l3*sin(psi - th)) - l2*sin(th)))
   -      ))/2. + (m2*(2*(-(l3*(psi' - th')*cos(psi - th)) -
   -         l2*th'*cos(th))*
   -       (-(l3*(psi' - th')*sin(psi - th)) + l2*th'*sin(th)) +
   -      2*(l3*cos(psi - th) - l2*cos(th))*
   -       (-(l3*(psi'' - th'')*cos(psi - th)) - l2*th''*cos(th) +
   -         l3*(psi' - th')**2*sin(psi - th) + l2*th'**2*sin(th)) +
   -      2*(-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th))*
   -       (l3*psi'*sin(psi - th) + th'*(-(l3*sin(psi - th)) - l2*sin(th)))
   -       + 2*(-(l3*sin(psi - th)) - l2*sin(th))*
   -       (l3*psi'*(psi' - th')*cos(psi - th) +
   -         th'*(-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th)) +
   -         l3*psi''*sin(psi - th) +
   -         th''*(-(l3*sin(psi - th)) - l2*sin(th)))))/2. +
   -  g*m1*(l1*sin(th) - l4*sin(phi + th)) -
   -  (m1*(2*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*
   -       (l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th)) +
   -      2*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*
   -       (-(l1*th'*sin(th)) + l4*(phi' + th')*sin(phi + th))))/2. +
```

```
-   (m1*(2*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*
-         (l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th)) +
-        2*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*
-         (-(l1*th'*sin(th)) + l4*(phi' + th')*sin(phi + th)) +
-        2*(l1*cos(th) - l4*cos(phi + th))*
-         (l1*th''*cos(th) - l4*(phi'' + th'')*cos(phi + th) -
-           l1*th'**2*sin(th) + l4*(phi' + th')**2*sin(phi + th)) +
-        2*(l1*sin(th) - l4*sin(phi + th))*
-         (l1*th'**2*cos(th) - l4*(phi' + th')**2*cos(phi + th) +
-           l1*th''*sin(th) - l4*(phi'' + th'')*sin(phi + th))))/2.= 0
```

Similarly, the equation for phi is

```
   -(g*l4*m1*sin(phi + th)) - (m1*
-      (2*l4*(phi' + th')*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*
-         sin(phi + th) - 2*l4*(phi' + th')*cos(phi + th)*
-         (l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th))))/2. +
-   (m1*(2*l4*(phi' + th')*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*
-         sin(phi + th) - 2*l4*(phi' + th')*cos(phi + th)*
-         (l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th)) -
-        2*l4*cos(phi + th)*(l1*th''*cos(th) - l4*(phi'' + th'')*cos(phi + th) -
-          l1*th'**2*sin(th) + l4*(phi' + th')**2*sin(phi + th)) -
-        2*l4*sin(phi + th)*(l1*th'**2*cos(th) - l4*(phi' + th')**2*cos(phi + th) +
-           l1*th''*sin(th) - l4*(phi'' + th'')*sin(phi + th))))/2.= 0
```

and the equation for psi is

```
    g*l3*m2*sin(psi - th) - (m2*(2*l3*(psi' - th')*
-         (-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th))*sin(psi - th) +
-        2*(l3*psi'*cos(psi - th) - l3*th'*cos(psi - th))*
-         (l3*psi'*sin(psi - th) + th'*(-(l3*sin(psi - th)) - l2*sin(th)))))/2. +
-   (m2*(2*l3*(psi' - th')*(-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th))*
-         sin(psi - th) - 2*l3*cos(psi - th)*
-         (-(l3*(psi'' - th'')*cos(psi - th)) - l2*th''*cos(th) +
-           l3*(psi' - th')**2*sin(psi - th) + l2*th'**2*sin(th)) +
-        2*l3*(psi' - th')*cos(psi - th)*
-         (l3*psi'*sin(psi - th) + th'*(-(l3*sin(psi - th)) - l2*sin(th))) +
-        2*l3*sin(psi - th)*(l3*psi'*(psi' - th')*cos(psi - th) +
-           th'*(-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th)) +
-           l3*psi''*sin(psi - th) + th''*(-(l3*sin(psi - th)) - l2*sin(th)))))/2.= 0
```

For the solution to the trebuchet's motion, as before, we solve these three differential equations simultaneously, subject to the initial conditions on the angles and their first derivatives. The results are shown in Appendix III.

Perhaps the most important result illustrated in this example is the local minimum that appears in the plot of θ vs. t. The beam "wobbles" a little bit because the projectile at the end of the sling has time to go through a pendulum-like oscillation, and pulls the beam with it. Not all values of the input parameters will yield this "wobble" in θ, but it is not uncommon. In this case, the maximum possible range occurs very close to the local minimum in θ, but it often appears elsewhere.

The efficiency of the trebuchet increases by the addition of the sling (of this particular length) from 59% (for the previous example) up to 81%. This is a substantial improvement, and indicates that trebuchets can be surprisingly efficient machines.

The position of the beam for maximum release is also later--it is 20.8° vs. 22.6° for the slingless example. One of the more interesting features of the range vs. time plot is that there is now a fairly broad plateau at the top--this model appears to be a little more forgiving in the release condition that gives a long range for the projectile.

The Trebuchet with a Sliding Sling

For a really realistic trebuchet model, we need to include the constraint on the sling that makes it slide without friction along the trough under the beam during the first part of the motion. The solution must be obtained in two parts. In the second part, the above three equations can be used for the motion during the unconstrained portion of the motion, after the projectile has left the slide, and before the projectile has left the sling. The initial conditions for this second phase come from the final conditions of the first part of the throw.

For even more realism, we could add a third phase to the throw: the motion of the trebuchet after the release of the projectile. This is of little interest at present, though it is easily done. We will stick with two phases and leave the three phase simulation as an exercise for the reader.

During the first portion of the movement, before the projectile has left the slide, we need to apply a suitable constraint to the motion. Refer to the diagram to derive the constraint:

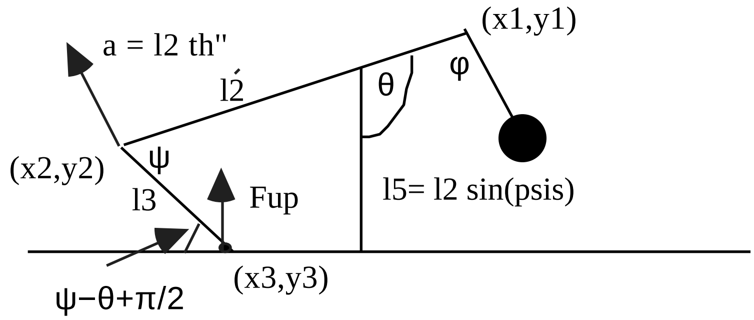


Fig. 7. The geometry of the trebuchet during the sliding phase of the sling.

We can calculate ψ when θ and φ, together with the starting angle for psi, ψs (= psis = $\pi/4$ in our example), are given. A little geometry shows that when the sling is in contact with the slide,

$$\psi = \theta - \pi/2 + \text{Arcsin}[(l2/l3)\ (\sin(\psi s)+\cos(\theta))]$$

Thus, only two simultaneous differential equations need to be solved during this phase of the movement: solving for θ as a function of time gives ψ immediately.

This phase of the movement ends when the force upward on the projectile (exerted by the sling) exceeds the force downward on it due to gravity. The upward force on the projectile comes from the tension, T, in the sling rope. A bit of geometry then shows that

$$m_3\ \ddot{x}_3 = T \cos(\pi/2+\psi-\theta)$$

and

$$m_3\ \ddot{y}_3 = T \sin(\pi/2+\psi-\theta)$$

Eliminating the unknown T between them yields

$$\ddot{y}_3 = \ddot{x}_3\ \tan(\pi/2+\psi-\theta) = g$$

as the condition for liftoff of the projectile from the slide. The time for liftoff, t = tlv, is when the projectile leaves the slide; at that point we can take the values of the angles and their derivatives as the initial condition for the second part of the motion. The method is to solve the two differential equations for θ and φ as functions of t, then (numerically) calculate tlv.

This then enables one to calculate the initial conditions for input to the three unconstrained differential equations derived above. The solution then proceeds as for the unconstrained model described above.

In the first phase of the movement, the Lagrangian equations are changed by the constraint. The method we use is given in many advanced textbooks in mechanics--the extension of Lagrange's equations to holonomic constraints. It can be carried out without too much difficulty as follows:

First, we put the constraint equation into a function

$$f(\theta,\varphi,\psi) = \psi - \theta + \pi/2 - \text{Arcsin}[(l2/l3)\ (\sin(\psi s)+\cos(\theta))]$$

and so we can verify that the constraint equation with parameters a_i satisfies the form

$$\text{eqconstr} = a_\theta\ \theta' + a_\varphi\ \varphi' + a_\psi\ \psi' = 0$$

where the ai are calculated by

$$a_\theta = (\partial\, f(\theta,\varphi,\psi)/\partial\theta)_{\varphi,\psi},$$

and similarly for a_φ and a_ψ.

We get $a_\varphi = 0$, $a_\psi = 1$, and

$$a_\theta = -1 + (l2/l3)\sin(\theta)\,\{1- ((l2/l3)\,(\cos(\theta) + \sin(\psi s)))^2\}^{-1/2}$$

One then uses for the Lagrange equations, involving the Lagrange multiplier λ,

$$\frac{d}{dt}\frac{\partial L}{\partial\dot\theta} - \frac{dL}{d\theta} - \lambda\, a_\theta = 0,$$

$$\frac{d}{dt}\frac{\partial L}{\partial\dot\varphi} - \frac{dL}{d\varphi} - \lambda\, a_\varphi = 0,$$

where now, since $a_\psi = 1$,

$$\lambda = \frac{d}{dt}\frac{\partial L}{\partial\dot\psi} - \frac{dL}{d\psi}.$$

Using these results, the differential equation for th is found to be

```
   ((l1**2 - l1*l2 + l2**2)*mb*th'')/3. +
-   (g*(l1 - l2)*mb*sin(th))/2. + g*m2*(-(l3*sin(psi - th)) - l2*sin(th)) -
-   (m2*(2*(-(l3*(psi' - th')*cos(psi - th)) - 2*th'*cos(th))*
-        (l3*(psi' - th')*sin(psi - th) - l2*th'*sin(th)) +
-         2*(-(l3*(psi' - th')*cos(psi - th)) -
-          l2*th'*cos(th))*(-(l3*(psi' - th')*sin(psi - th)) + l2*th'*sin(th))))/2. +
-   (m2*(2*(-(l3*(psi' - th')*cos(psi - th)) -
-          l2*th'*cos(th))*(l3*(psi' - th')*sin(psi - th) -
-          l2*th'*sin(th)) +  2*(-(l3*(psi' - th')*cos(psi - th)) - l2*th'*cos(th))*
-         (-(l3*(psi' - th')*sin(psi - th)) + l2*th'*sin(th)) +
-          2*(l3*cos(psi - th) - l2*cos(th))*
-          (-(l3*(psi'' - th'')*cos(psi - th)) - l2*th''*cos(th) +
-          l3*(psi' - th')**2*sin(psi - th) +
-          l2*th'**2*sin(th)) + 2*(-(l3*sin(psi - th)) - l2*sin(th))*
-        (l3*(psi' - th')**2*cos(psi - th) - l2*th'**2*cos(th) +
-          l3*(psi'' - th'')*sin(psi - th) -
-          l2*th''*sin(th))))/2. - (-1 + (l2*sin(th))/(l3*Sqrt(1 -
-          (l2**2*(cos(th) + sin(psis))**2)/l3**2)))*
-    (g*l3*m2*sin(psi - th) - (m2*(2*l3*(psi' - th')*(-(l3*(psi' - th')*cos(psi - th)) -
-             l2*th'*cos(th))*sin(psi - th) +  2*l3*(psi' - th')*
-         cos(psi - th)*(l3*(psi' - th')*sin(psi - th) -
-             l2*th'*sin(th))))/2. + (m2*(2*l3*(psi' - th')*
```

```
-             (-(l3*(psi' - th')*cos(psi - th)) -
-             l2*th'*cos(th))*sin(psi - th) + 2*l3*(psi' - th')*
-             cos(psi - th)*(l3*(psi' - th')*sin(psi - th) -
-             l2*th'*sin(th)) -  2*l3*cos(psi - th)*(-(l3*(psi'' - th'')*cos(psi - th)) -
-             l2*th''*cos(th) + l3*(psi' - th')**2*sin(psi - th) + l2*th'**2*sin(th)) +
-           2*l3*sin(psi - th)*(l3*(psi' - th')**2*cos(psi - th) - l2*th'**2*cos(th) +
-             l3*(psi'' - th'')*sin(psi - th) - l2*th''*sin(th))))/2.) +
-  g*m1*(l1*sin(th) - l4*sin(phi + th)) - (m1*(2*(l1*th'*cos(th) -
-        l4*(phi' + th')*cos(phi + th))*(l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th)) +
-      2*(l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*(-(l1*th'*sin(th)) +
-         l4*(phi' + th')*sin(phi + th))))/2. + (m1*(2*(l1*th'*cos(th) -
-        l4*(phi' + th')*cos(phi + th))*(l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th)) +
-      2*(l1*th'*cos(th) -  l4*(phi' + th')*cos(phi + th))*(-(l1*th'*sin(th)) +
-         l4*(phi' + th')*sin(phi + th)) + 2*(l1*cos(th) - l4*cos(phi + th))*
-       (l1*th''*cos(th) - l4*(phi'' + th'')*cos(phi + th) - l1*th'**2*sin(th) +
-         l4*(phi' + th')**2*sin(phi + th)) + 2*(l1*sin(th) - l4*sin(phi + th))*
-       (l1*th'**2*cos(th) - l4*(phi' + th')**2*cos(phi + th) +
-         l1*th''*sin(th) - l4*(phi'' + th'')*sin(phi + th))))/2..eq.0
```

and for the equation for phi is unchanged from before because $a_\varphi = 0$:

```
    -(g*l4*m1*sin(phi + th)) - (m1*(2*l4*(phi' + th')*(l1*th'*cos(th) -
-         l4*(phi' + th')*cos(phi + th))*sin(phi + th)- 2*l4*(phi' + th')*cos(phi + th)*
-       (l1*th'*sin(th) - l4*(phi' + th')*sin(phi + th))))/2. + (m1*(2*l4*(phi' + th')*
-       (l1*th'*cos(th) - l4*(phi' + th')*cos(phi + th))*sin(phi + th)
-        - 2*l4*(phi' + th')*cos(phi + th)*(l1*th'*sin(th) -
-         l4*(phi' + th')*sin(phi + th)) - 2*l4*cos(phi + th)*(l1*th''*cos(th) -
-         l4*(phi'' + th'')*cos(phi + th) - l1*th'**2*sin(th) + l4*(phi' + th')**2*
-         sin(phi + th)) - 2*l4*sin(phi + th)*(l1*th'**2*cos(th) -
-        l4*(phi' + th')**2*cos(phi + th) +
-         l1*th''*sin(th) - l4*(phi'' + th'')*sin(phi + th))))/2..eq.0
```

When using Mathematica, these two equations, together with the initial conditions, are sufficient input into the differential equation solver to get the values of the three angles and their first derivatives as a function of time, which then yield all of the desired characteristics of the throw.

When using other languages, one needs to choose and set up another differential equation solver. Fortunately, the differential equations are not stiff, so a simple fourth order Runge-Kutta algorithm suffices. In most of these methods, the equations for th, phi, and psi given above need to be solved for their second derivatives, each of which are put into functions called by the Runge-Kutta subroutine. These were solved for using Mathematica and are given in Appendix IV.

The solution to the sliding sling model is fairly complex, and will not be described in all of its details here. We use the same values as previously: m1=100, m2=1, mb=0, 11=1, l2=4, l3=3.25, l4=1, l5=4/sqrt[2]; θ[0]=135°, φ[0]=45°, and ψ[0]=45°. The results are quite similar to those for the unconstrained model. The range was 282.4 ft, giving an efficiency of 83% compared to the previous 81%. It would leave when θ = 18.4° at a time 0.40 s after the release.

A Nominal Trebuchet

The results for a related design in which the mass of the beam is a more reasonable 5 times the mass of the projectile are shown in Appendix V. This particular design was used for debugging the various implementations in other languages, and since the Mathematica language is less accessible to many readers of this document, they are included here.

In particular, the first set of graphs show the partition of energy into the parts of the treb as a function of time. The flatness of the total energy, shown in the very first graph, is a very important verification of the model--it shows that the energy is conserved in the solution, as it should be. And there is no jog at the point where the projectile leaves the slide, showing the constraint was properly done.

The graph for the kinetic energy in the CW is also interesting. It shows that at one point it is very low, nearly zero, implying that it is "stalled." The beam stalls a little later. This implies a relatively efficient design.

The next graph is a parametric representation of the x-y positions of the projectile and the center of mass of the counterweight. The two dots are drawn at the time that the projectile would leave in order to attain the maximum range. Note that it is very close to the time that the CW abruptly changes direction at the bottom of its travel. This change shows how close the release of the projectile is to the "jerk" in the counterweight travel. When it is very close, like this one is, the error in the simulation is probably greater because of the increased error introduced by the approximations of assuming all the parts are perfectly rigid.

The two graphs for the calculated range show that there is only a single maximum, and that no complications due to the possible multi-valuedness of the range vs. psi occurs, making this one a "nicer" design.

The graphs of the accelerations experienced by the projectile and counterweights are shown next. They show a very slight jog at the time the projectile comes off the slide, implying that the numerical "match-up" condition is not perfect for some obscure reason. It is sufficiently small, though, to not detract from our confidence in the model. The accelerations are given in m/s^2, and are seen to be quite high (g is 9.8 m/s^2). These are surely overestimates of that actually experienced in a real trebuchet because of the assumed perfect rigidness of all of the parts.

The appendix also shows for completeness the values obtained by the Mathematica Program (Trebfornet5.04) for various other characteristics of the throw. These are pretty much self-explanatory. One of particular interest (if you want to throw a cow or something) gives the acceleration on the projectile just before release.

The effect of added mass in the beam

All of the models discussed up to now used mb = 0. The effect of mass in the beam would be expected to decrease the efficiency of the trebuchet, so we need to explore how much it is affected and look for other effects. A run with mb =5 was made (using the unconstrained sling model) and the plot shows the result for θ[t] compared with that for mb=0. The mass in the beam has smoothed out the motion so that no minimum occurred. The efficiency dropped from the 83% obtained for mb=0 down to 67%.

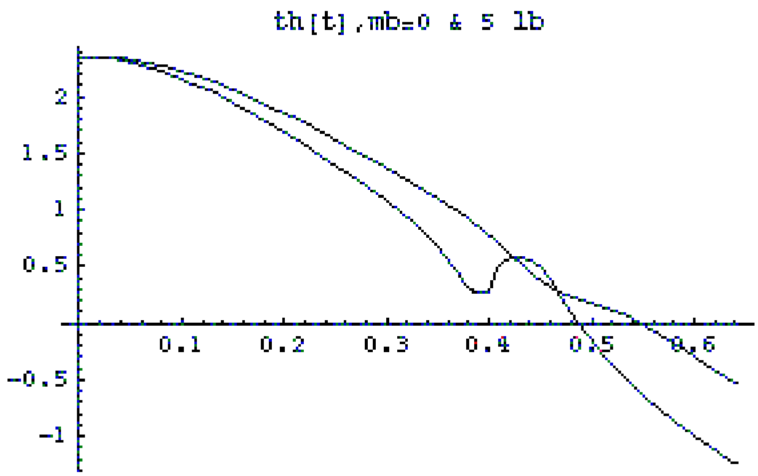


Fig. 8. A comparison of the θ(t) solutions for trebuchets with mass zero amd mb=5 compared. The mb=0 case has a minimum at time 0.4.

A study of the dependence of the efficiency of the trebuchet using various masses for the beam can also be easily carried out. For the free sling case, we get the results shown in the figure. Increasing the mass of the beam decreases the efficiency quite dramatically. A good design will have a beam as light as possible. In all of the models considered so far we used the simplest assumption of a uniform cross-section of the beam. Other more complicated shapes could be easily incorporated in the model. One needs to calculate the moment of inertia about the axle and the distance of the center of mass from the axle.

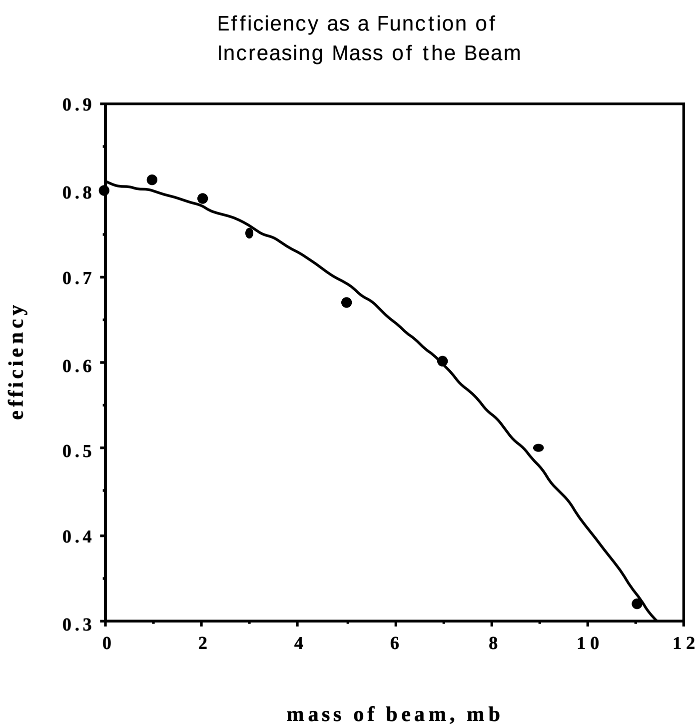


Fig. 9. The effect of the mass of the beam on the efficiency of the trebuchet.

Summary of the models so far

We can summarize the results obtained up to now in Table I.

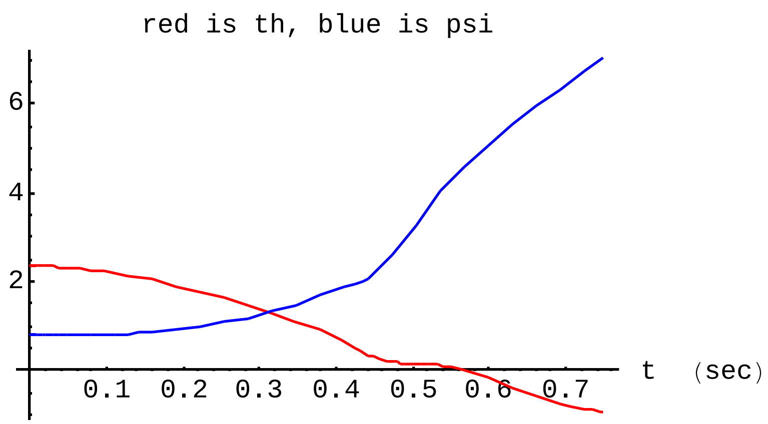
Table I. A comparison of the results for the models discussed. They all have a CW of 100 lb, projectile of 1 lb, short arm of 1 ft, long arm of 4 ft. The + hcw model has a hinged cw 1 ft long, the + sling, +sliding and + mb models also include a sling 3.25 ft long.

	see-saw	**+hcw**	**+sling**	**+sliding**	**+ mb = 5**
R	38.4 ft	203	276	284	228
efficiency	11%	59	81	83	67
θ release	38.0°	22.6	20.8	18.4	19.3
t release	0.40 s	0.36	0.40	0.40	0.46
Minimum in θ?	no	no	yes	yes	no

Modes of Behavior

The various modes of behavior of the sling in relation to the beam are of particular interest in interpreting and understanding the trebuchet. Consider first, a model similar to that shown before with the parameters m1=100, m2=1, mb=5, l1=1,l2=4, l3=3.25, and l4=1. We use the unconstrained sling model, with the usual initial conditions: the beam is at 45° with respect to the horizontal, and the counterweight hangs straight down. The behavior of θ, ψ and the range are shown in Fig. 10.

These parameters are chosen so that "good" throwing behavior results. The resulting angles are both a monotonic function of time. The maximum range of 235 ft occurs when the release occurs at 0.465 s. This happens when the beam is almost vertical (θ=13°). The angle between the beam and the sling is then 145.6°. There is a second smaller maximum for the range after the global maximum, which occurs because the sling continues to swing around (ψ keeps growing).



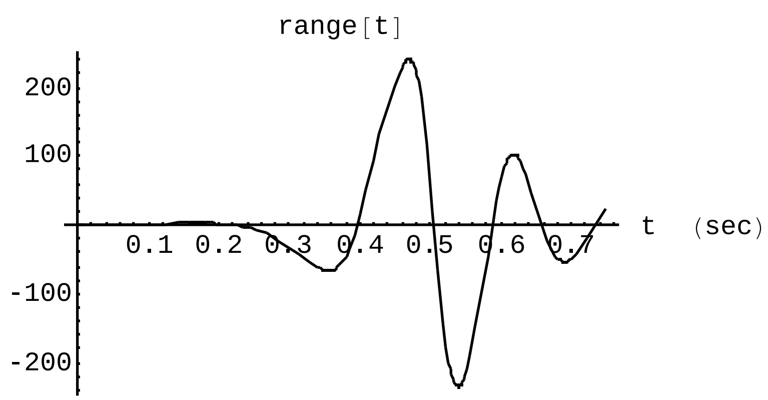


Fig. 10. The behavior of the trebuchet for a favorable set of parameters.

A somewhat more complicated movement occurs when the sling is shortened to 2 ft. as shown in Fig. 11. Now there is a small minimum in ψ that occurs in the vicinity of the maximum in the range.

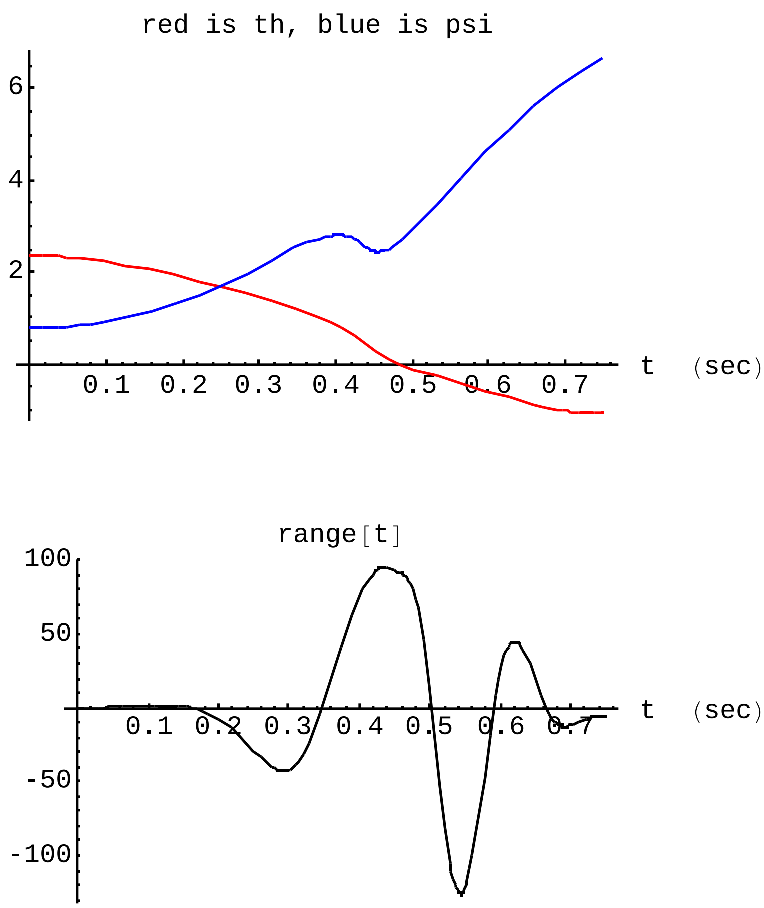
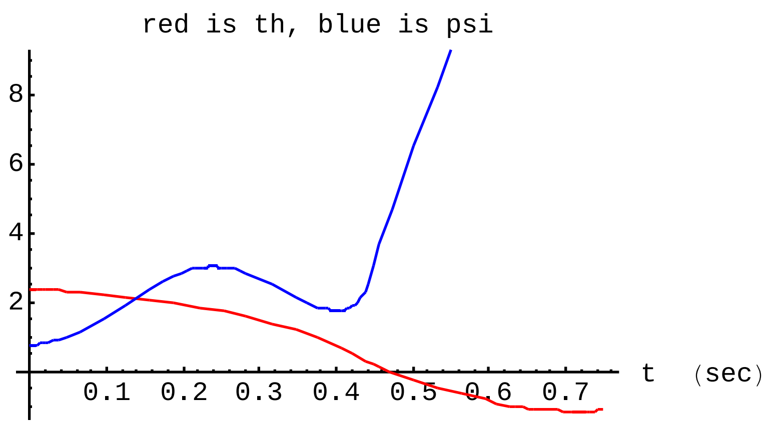


Fig. 12. The behavior of the trebuchet for a shortened sling.

Still more complex behavior ensues when the sling is shortened to 0.5 ft, as shown in Fig. 13. Here the sling and projectile get into a "twirling" mode--ψ keeps increasing monotonically (up to about 6 π, or three circles) at about the time the beam is vertical. Before this, there is a single broad maximum in ψ.



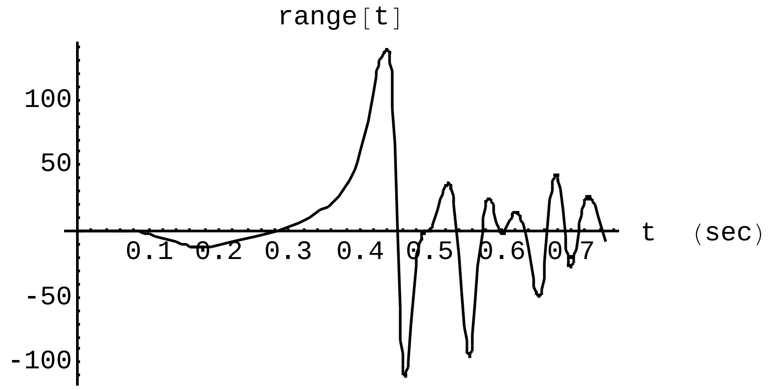


Fig. 13. The behavior when the sling is very short (0.5 ft in the example).

Evidently the behavior of the trebuchet can be quite complex.

The sling release mechanism

The mechanism that is used to release the sling is critical, of course, to the proper and efficient functioning of the trebuchet. The previous discussion of the models just assumed that we could get the release to occur at the proper moment. We have pointed out, however, that simple mechanisms would release at the first occurrence of some preset angle between the sling and the beam-- mainly because the mechanisms commonly used are not able to do otherwise. Not impossible to devise, however...

The usual arrangement for the release is to have a fixed projection from the beam (a hook, prong or a peg) over which a loop of the sling or a ring tied to one end of the sling slides freely. We will refer to the peg as a the "finger". The other end of the sling is permanently attached to the beam so it does not go flying off with the projectile. When the beam is horizontal and not moving, it would look something like that shown in Fig. 14.

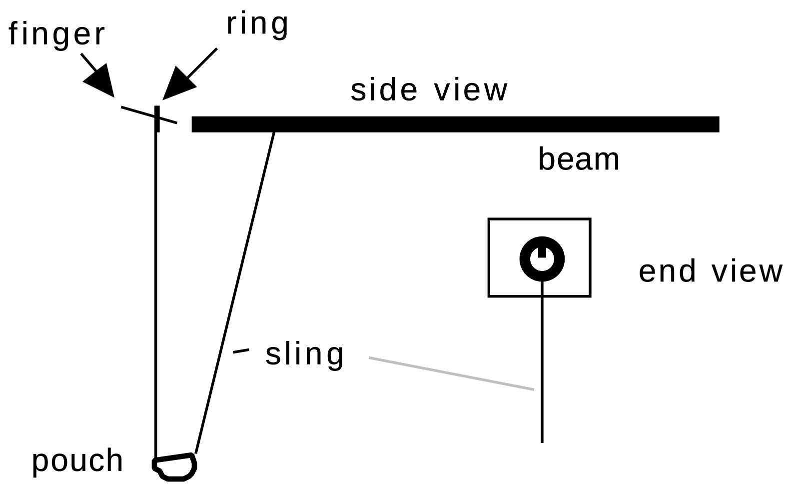


Fig. 14. The sling release mechanism in side and end views.

Although some designs rely on a loop of the sling slipped over the finger, it is probably more reproducible to rely on a ring of metal, such as a metal washer, that is fastened to one end of the sling. The finger can be mounted on the top surface of the beam by some means, or for smaller trebuchets it can be simply screwed into the end of the beam. One can use a clothesline type of hook that is straightened out, for example.

In general the finger is placed at some angle with respect to the beam. It is useful to define this angle, δ, with respect to the extension of the beam (so that it is an acute angle). A closer view of this is shown in Fig. 15.

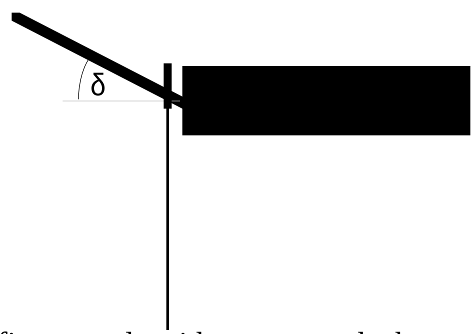


Fig. 15. Definition of the finger angle with respect to the beam.

First of all, it should be clear that in the absence of any friction, the ring will start to slide as soon as the angle between the finger and the sling decreases below 90°. In the limiting case in which the finger is very short, and friction is neglected, one could set the finger angle to be $\delta = \psi_r - 90°$, where ψ_r is the sling/beam angle given as the optimal release condition from the simulation. When friction cannot be neglected, the release will be sometime after the sling/finger angle, is greater than 90°, and δ should be set to a <u>smaller</u> angle than this.

Very often, the simulations give an optimal release angle of about $\psi_r = 160°$, so to this approximation, one would choose $\delta = 70°$.

The influence of the static force of friction between the ring and the finger is rather easy to be taken into account, and is worth looking at, and can give a better approximation of what value δ should be set to.

The configuration of the sling during the motion of the beam, with the angle ψ, used previously, and its relationship with δ is shown in Fig. 16.

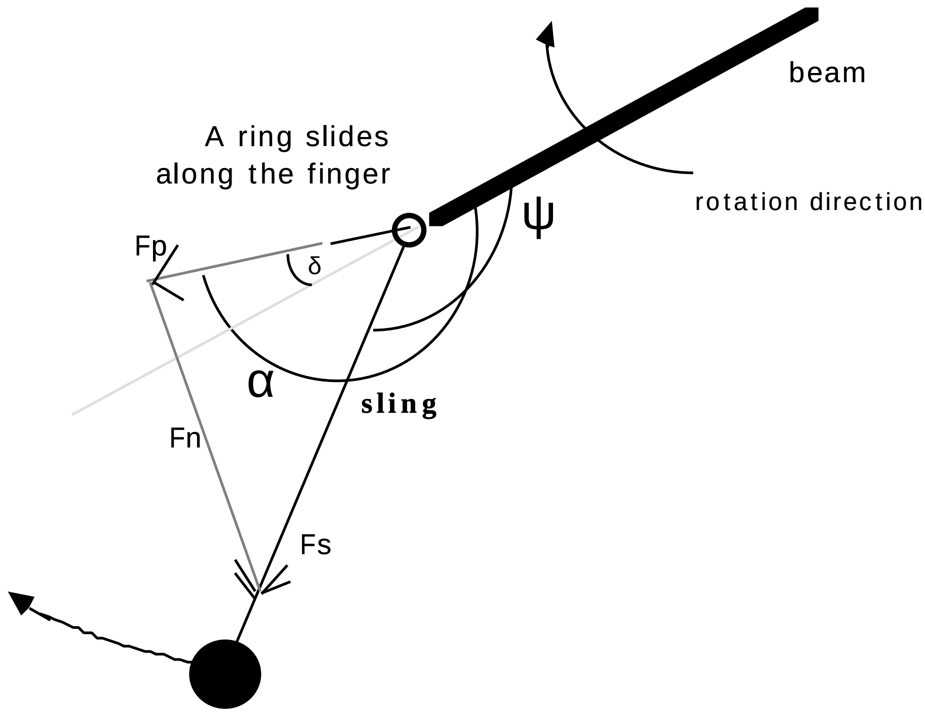


Fig. 16. The configuration of the sling during the throw, before release, showing the definition of the angles and forces involved.

Since the sling is made out of string or rope, it can only exert a tension force along itself, in the direction shown--the rotating projectile exerts a force, Fs, on the ring, tending to pull the ring along the finger. In principle, this force can be obtained from the solution of the differential equations given above, and depends on $\psi(t)$. It can be resolved into a force normal to the finger, F_n, and one along the finger, F_p.

The force tending to hold the ring fixed relative to the finger, resisting the pull F_p, is the force of friction, $F_f = \mu$ Fn, where μ is the static coefficient of friction between the ring and the finger. From the geometry shown, it can be seen that

$$F_f = \mu \, F_n = \mu \, F_s \sin(\alpha - \psi)$$

where

$$\alpha = \delta + \pi$$

The force exerted along the finger by the projectile, through the sling, is

$$F_p = F_s \cos(\alpha - \psi).$$

The ring just begins to slide when $F_p = F_f$. That is,

$$F_s \cos(\alpha - \psi) = \mu F_s \sin(\alpha - \psi)$$
or
$$\text{when } \cot(\alpha - \psi) = \mu$$
or
$$\cot(\pi + \delta - \psi) = \mu$$

Which leads to the final result, that the ring will begin to slide when

$$\psi = \pi + \delta - \text{arc cot } \mu,$$

where, of course, the angles are expressed in radians.

If we are content to assume that the length of the finger is close enough to zero and therefore only need to take the static force of friction into account (i.e., release occurs at the instant that the ring starts to slide), then we should set the finger at an angle

$$\delta = \psi_r - \pi + \text{arc cot}(\mu).$$

Again, when $\psi_r = 160°$ and $\mu = 0$, this gives $\delta = 160° - \pi + \text{arc cot } (0) = 70°$, which checks with the result given above.

For clean steel $\mu = 0.58$, and for lubricated steel, $\mu = 0.1$, so the corresponding angles (*i.e.*, arc cot μ) are then 60° and 84°, respectively, instead of 90° when μ=0. For the nominal case of $\psi_r = 160°$, therefore, we would therefore set the finger angle at 40° or 64°, for clean or lubricated steel, respectively.

The condition for sliding is, unfortunately, not the same as the condition for release of the sling, except in the limiting case where the length of the finger approaches zero. For a finger having a finite length, the differential equation for sliding with a (variable) force coming from the sling acting on the ring, opposed by the sliding friction force, would have to be solved. Since the force varies with time during the sliding, another (simultaneous) differential equation would be added to the ones already described. While this is doable, it is perhaps more complex than necessary.

First, note that the sliding phase of the ring delays the release of the sling and the projectile. This means that we would need to set the finger to a smaller angle than that calculated above. If we know how long the slide takes, and the average rate that the sling/beam angle is changing, we can readily calculate how much to correct the finger angle.

The simulation can easily provide the acceleration of the projectile at the instant that sliding starts. This information can provide us with the tension in the sling, and therefore the pulling force and the acceleration of the ring. If we make the approximation that this acceleration is a constant, a_r, then the time interval during which the ring slides is simply

$$t_s \approx \sqrt{\frac{2\, l_f}{a_r}},$$

where l_f is the length of the finger. To the same approximation, therefore, the change in ψ during the sliding is

$$\psi_s \approx t_s\, \dot{\psi}_s\ ,$$

and we would therefore set the finger angle to

$$\delta \approx \psi_r - \pi + \text{arc cot}(\mu) - \dot{\psi}_r \sqrt{2\, l_f/a_r}\ .$$

The effective acceleration along the finger is readily seen to be

$$a_r = a_p\, [\cos(\pi+\delta-\psi) - \mu_k \sin(\pi+\delta-\psi)]$$

where μ_k is the kinetic coefficient of friction, a_p is the acceleration of the projectile when $\psi = \psi_r$.

These last two equations can be solved numerically for the desired δ when the coefficients of friction, the finger length, the release angle and its rate of change are all known. An easy method is to plot the lhs of the equation

$$\psi_r - \pi + \text{arc cot}(\mu) - \dot{\psi}_r \sqrt{2\, l_f}/\sqrt{(a_p[\cos(\pi+\delta-\psi)-\mu_k \sin(\pi+\delta-\psi)])} - \delta = 0$$

as a function of δ and find the zero. Alternatively, the FindRoot function in Mathematica can be used:

```
fingsol[del_,psir_,lf_,ap_,mu_,muk_,psidot_]:=
  psir-Pi+ArcCot[mu]-
    psidot*Sqrt[2*lf/(ap*(Cos[Pi+del-psir]-muk*Sin[Pi+del-psir]))]-del;
fingerangle=del/.FindRoot[fingsol[del,Pi*160/180,.1,280,.4,.1,18],{del,0,1}].
```

As an example we use the base case model, which has
l1=1 ft, l2=4, l3=3.5, l4=1, m1=100 lb, m2=1, mb=5. We assume for our model that $\mu = 0.4$ and $\mu_k = 0.1$ and the finger length $l_f = 10$ cm, and we get the values $\dot{\psi}_r = 18$ rad s^{-1}, and $a_r = 280$ m s^{-2}. The graph produced is shown in Fig. 17, and gives a value for δ of 0.284 rad = 16.3°. Thus the sliding correction is not negligible.

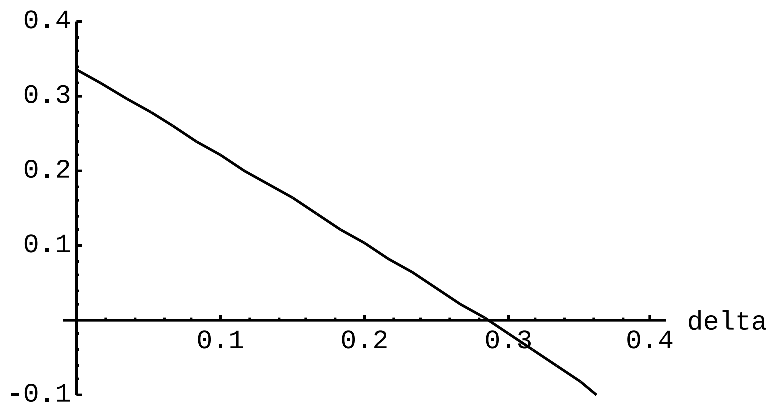


Fig. 17. Solution for the finger angle for the nominal case.

The coefficient of static friction is can vary quite a bit for different degrees of cleanliness of the surfaces. The value of 0.58 given above is an upper limit. Some texts give numbers as low as 0.15, and a coefficient of kinetic friction of 0.09. The static coefficient of friction can be readily determined for a real finger/ring arrangement. Put the ring on the finger while the finger is horizontal. Then tilt it till it starts to slide. The angle of the finger with respect to the horizontal, s, is then measured. The coefficient of friction is then μ = arc tan(s).

To summarize, we can make a fairly good calculation of where to set the finger angle, and can readily calculate it without having to solve any additional differential equations. However, there are several approximations involved, and there is some room for error in measuring the required friction coefficients, so the results should perhaps be taken not too literally. However, it would certainly appear from limited experience with the simulation and real models, that a finger angle of 20° is generally not too bad a place to start.

Mathematical Correctness of the Model

All of models described, but especially the constrained sling model, are sufficiently complex that one needs to carefully check them for correctness. A number of checks can be used. One check that can be done involves the dimensional correctness--does a model with lengths and masses doubled give twice the range? The models pass this test. Another involves the "reasonableness" of the movement. Does it "look" right? Animations of the solutions of both the constrained and unconstrained cases seem to be impressively natural. Another check involves using the solution for the variables and their derivatives to calculate the potential and kinetic energy of all of the parts as a function of time. They should sum to a constant, independent of time. The models satisfy this check too.

Other tests would involve the behavior at limiting conditions, such as setting m1, m2 and mb so that they very nearly balance when the beam is horizontal. It should move very slowly in the direction expected. The models satisfy this check. The same model written up in two different languages on different platforms should also give results in good agreement with one another. This has been done for both the constrained and unconstrained model. Taken together, all of these observations point to the conclusion that the models are correct with a high degree of confidence.

Optimal Design Considerations

The mathematical model can be used to aid in the design of real trebuchets. However, since there are 7 parameters and three initial conditions, the parameter space one can explore is quite large. One help is to realize that one might as well only work with "normalized" parameters by dividing all of the lengths by l1, and the masses by m2, say. This reduces the problem to 5 parameters and three initial conditions, but still the parameter space is large, and exploration requires a fast simulator.

One method is to use the "free sling" model in a do-loop in which one variable at a time is varied. Plots of the efficiency are perhaps the most useful. Figs. 18 and 19 show such plots generated around the nominal parameters used up to now: l1=1, l2=4, l3=3.5, l4=1, m1=100, m2=1, mb=5. Evidently, from these plots a sling length of 3.5 and counterweight mass of 100 are good. Anything less than these values results in a fall-off of efficiency.

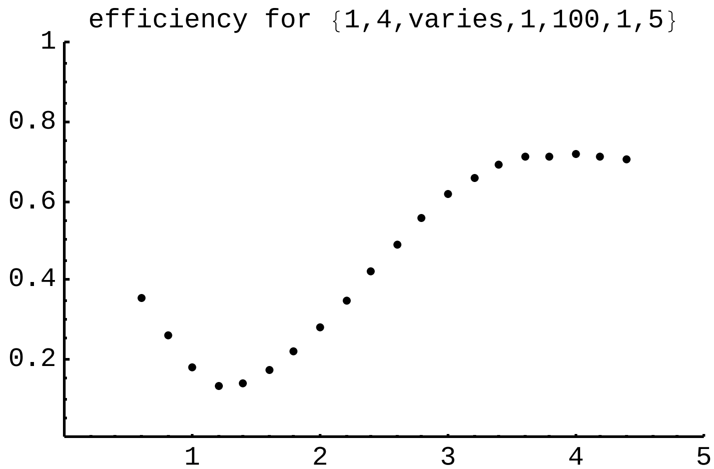


Fig. 18. The efficiency of the nominal trebuchet as the length of the sling varies.

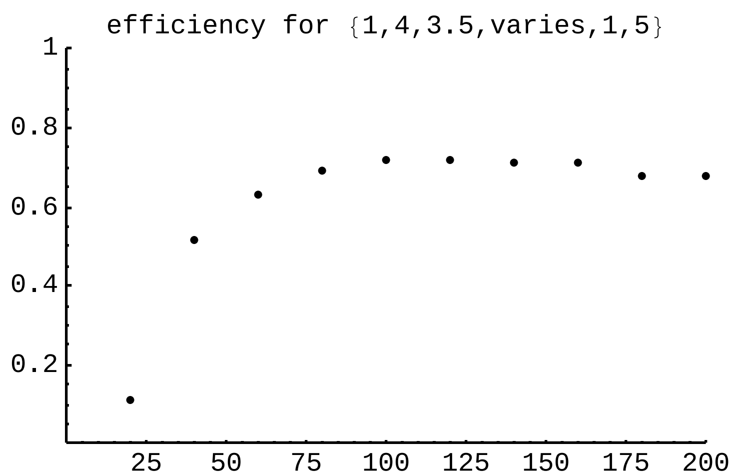


Fig.19. The efficiency as a function of the counterweight mass.

Effect of Propped Counterweights

The effect of propping the counterweight (making the starting CW/beam angle, phiz, something other than that for a freely hanging one) has been the subject of some speculation. Propping the counterweight necessarily makes its center of mass higher, which, one would think, would most likely make the range greater--the distance the CW can fall is greater. Its effect on the range efficiency is not so clear because propping the weight will induce a greater swinging tendency, leading to a lowered efficiency. However, perhaps the induced swinging could counteract the "natural" swing, leading to increased efficiency. This can only be settled by experiments or simulations--our physical intuition is just not really good enough.

Figs. 20 a and b show the results of two such simulations. One simulation (solid circles) used the set
{ l,m} = {4,16,12, 2; 400, 2, 20} and the other (open circles) used { l,m} = {1, 4, 3, 1; 100, 1, 5}, and the beam of both is at 45° with respect to the ground. The simulations were both carried out for a constrained, sliding sling model in which the "forbidden zone" behavior of the sling release is ignored. They used MacTreb 2.0.

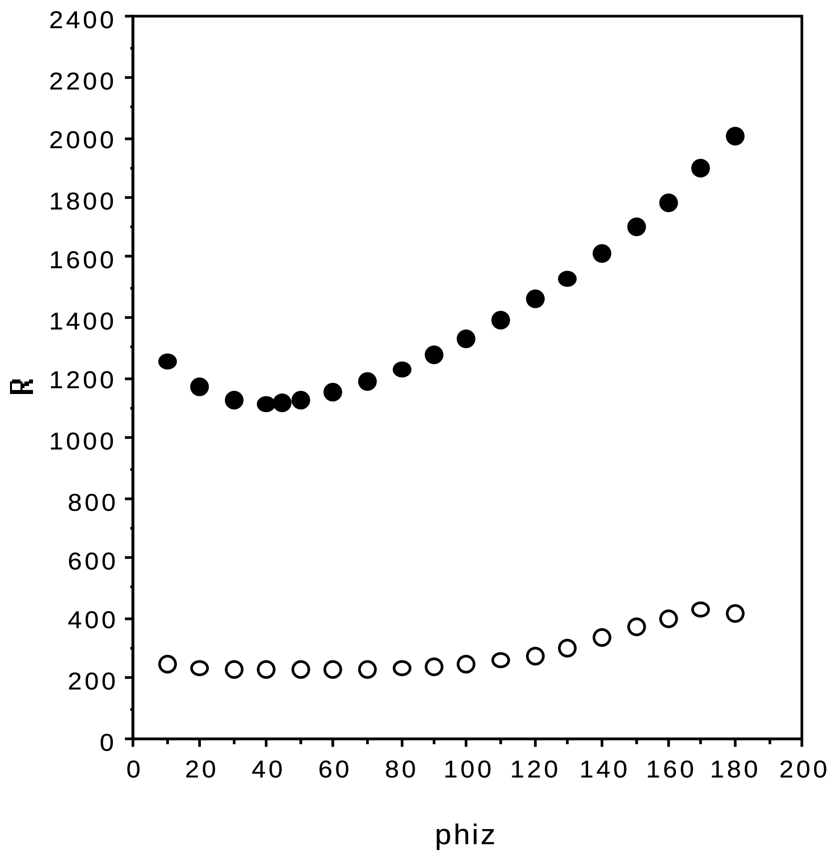


Fig. 20a The range for two trebuchets:

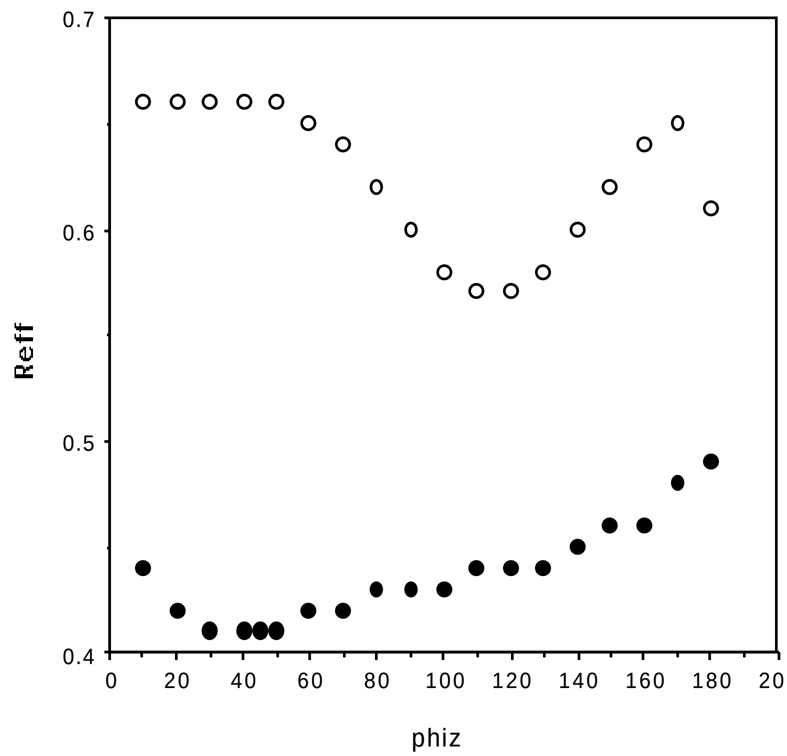


Fig. 20b. Range efficiency for the two trebuchets in Fig. 20a.

It is clear from these two graphs that it is difficult to generalize. For one, the range is hardly affected over a considerable span of propping angle because an efficiency decrease nearly cancels the increase in the distance the counterweight falls. In the other, less optimized treb, the range and range efficiency increase at the same time for propping angles above 45°. Evidently one considering constructing a propped CW design should carry out a few simulations before-hand.

It is not clear from these graphs how much the shock on the machine during the course of the throw would change by propping either. This can be estimated for the models, but require some extra calculations. However, any decrease in the range efficiency such as that shown for the smaller trebuchet, should make one hesitate to prop it to an angle of 100°, for example. Less efficient trebs surely are more at risk of shaking apart and/or breaking the axle.

Monte Carlo Design

Another method of exploring the rather large parameter space is to use a "Monte Carlo" approach, which involves calculating the results for randomly chosen parameters within some region of the design space.

For this method to work, we need a faster method for calculating the results, and a Fortran program, MonteCarloTreb1.f was devised. This program can be downloaded from the

"Algorithmic Beauty of Trebuchets" site. It uses a Runge-Kutta method with variable step size, and is considerably faster than the previously published fortrtan program freeslcw3.0. It does about 5 throws per second on a Power Mac 8500/180.

Since we are interested in allowing many of the parameters to vary, one needs a basis for comparison of the goodness of different designs. Obviously, a treb with a longer beam and heavier counterweight will throw farther, so some other basis for comparison should be used. Range efficiency is, of course the desired quantity, but we must further work with models that have the same input energy. We will therefore keep constant the distance that the counterweight can fall, and its mass a constant too.

For this example, we will also keep the projectile mass and the mass of the beam a constant, and only optimize on the lengths. We choose, based partly on the results shown above, to make m1=100, m2=1, and mb=5 lbs. We use the "free sling" model, which has the sling initially horizontal, as usual, and the CW hanging vertically.

We want to let the angle of the beam with respect to the ground, psiz, vary between 30° and 60°. These angles correspond to a shallow and steep pitch of the beam. Fixing psi fixes the other angles as thz=psiz+π/2, and phiz = π - th = psi+π/2. The distance the counterweight falls, h, is given by h= l1 * (1 + sin(psiz)) and is assumed to be a constant, so l1 is determined from this to be l1 = h/(1 + sin(psiz)).

An important design parameter is the ratio of the lengths of the long arm to the short arm of the beam, l2/l1. Most published designs and drawings appear to have this ratio between 3 and 5, so we choose to explore this region as well. Given the constraint on l1 just given, choosing a ratio randomly from within this range gives l2. Then the height of the axle is fixed as well at l5 = l2*sin(psiz).

For the length of the sling, we will look in the region where the sling length is about equal to the length of the long arm of the beam: l3/l2 = .5 to 1.5. For the length of the CW suspension, we choose, rather arbitrarily to look at l4/l1= 0.5 to 1.0. This cannot be made to be too large--else the counterweight will crash into the ground.

To summarize, the space being explored has 4 dimensions: psiz, l2/l1, l3/l2, and l4/l2. all of the other lengths and angles being determined from them. And to reiterate, we set m1=100, m2=1, and mb=5 lbs. Psiz varies from 30° to 60°, and we assume the CW falls a constant distance for all of the proposed designs. We set this distance, rather arbitrarily to be h = 2 ft. This is Stage 1 of our exploration.

The Monte Carlo program was run for 200 trials chosen randomly within this 4-dimensional space, and the range efficiency determined for each. There are therefore four graphs. Fig. 20 shows the results as plots of the range efficiency vs. each of the varied parameters. Here we can see that the range efficiency for this collection of trebs ranged from about 0.1 up to about 0.7. Thus, guessing a single design within the bounds set above, could lead to a very inefficient treb. It is better to design, than guess.

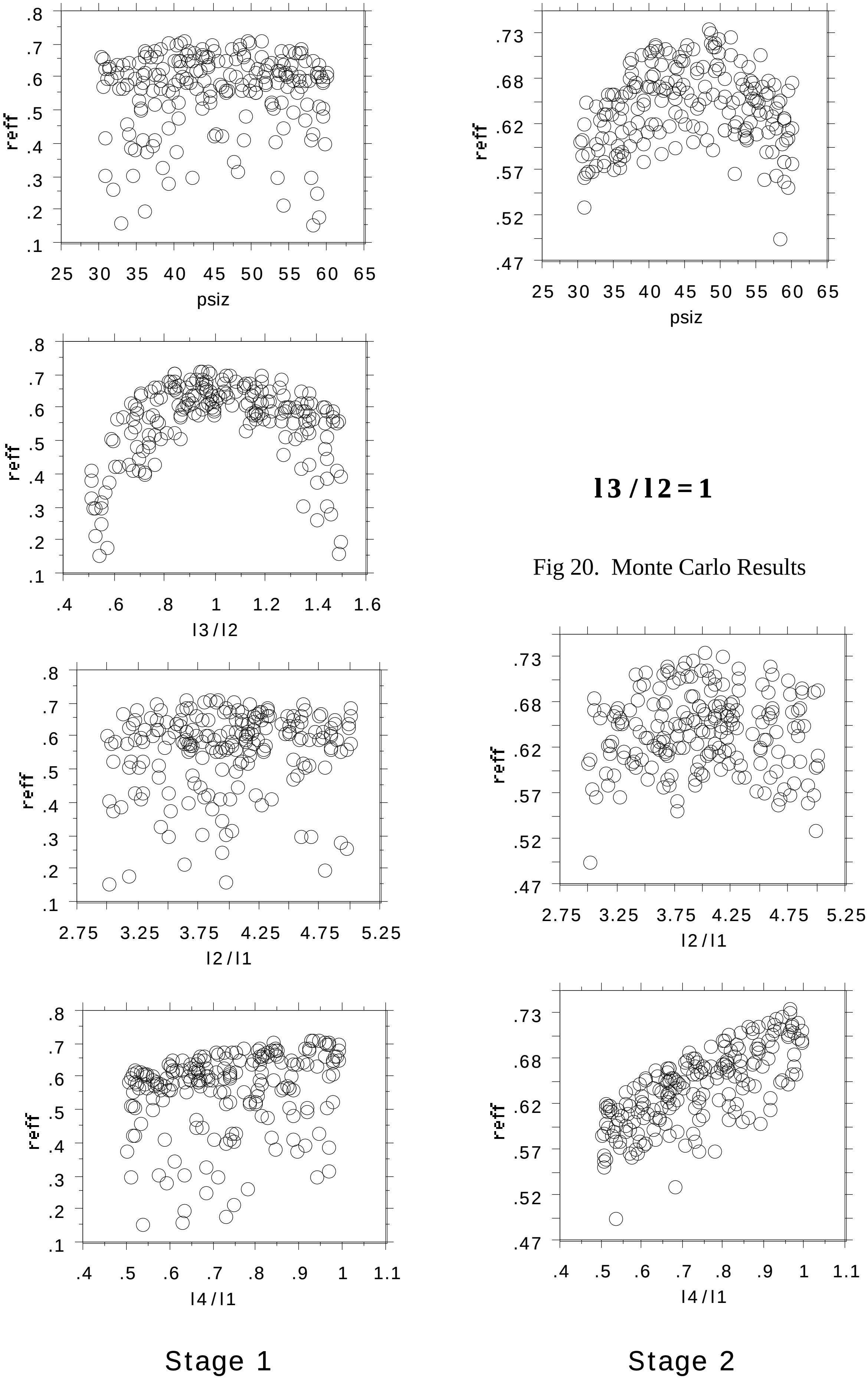


Fig 20. Monte Carlo Results

The first and clearest conclusion reached from these four graphs is that the length of the sling appears to have an optimum at about l3/l2=1. The range efficiency varies greatly for the other parameters, with a slight trend toward larger values of l4.

A little clearer picture emerges if we impose a further constraint: a second stage of 200 trials was therefore carried out with l3/l2 now constrained to be unity. This is also shown in Fig. 20. Now a fairly clear optimum in psiz appears--it is close to 45°. Again, the trend toward higher efficiency for larger values of l4 is apparent. Making l3=l2 has decreased the range of efficiencies very substantially. It is now between 0.50 and 0.73.

The upward trend of efficiency with the length of the CW arm, l4, may be readily explained: the longer this arm, the less motion in the x-direction that the CW experiences during its fall, and therefore the less wasted energy in the CW at the time of release. A very long counterweight arm, however, is not permitted in the usual design of a trebuchet: the counterweight cannot be allowed to hit the ground or the slide for the projectile. An efficient design would have l4 be such that the bottom of the CW just misses hitting the ground. If we say that the center of mass of the counterweight is in the center of the counterweight, this means that we must have (as a little geometry shows)

$$l4 < (l1/2) * (l2 * \sin(psiz) - l1)/l1.$$

The results so far indicate that there is comparatively little impact of the choice of the long arm to short arm of the beam (l2/l1) on the efficiency. Only a shallow optimum is seen.

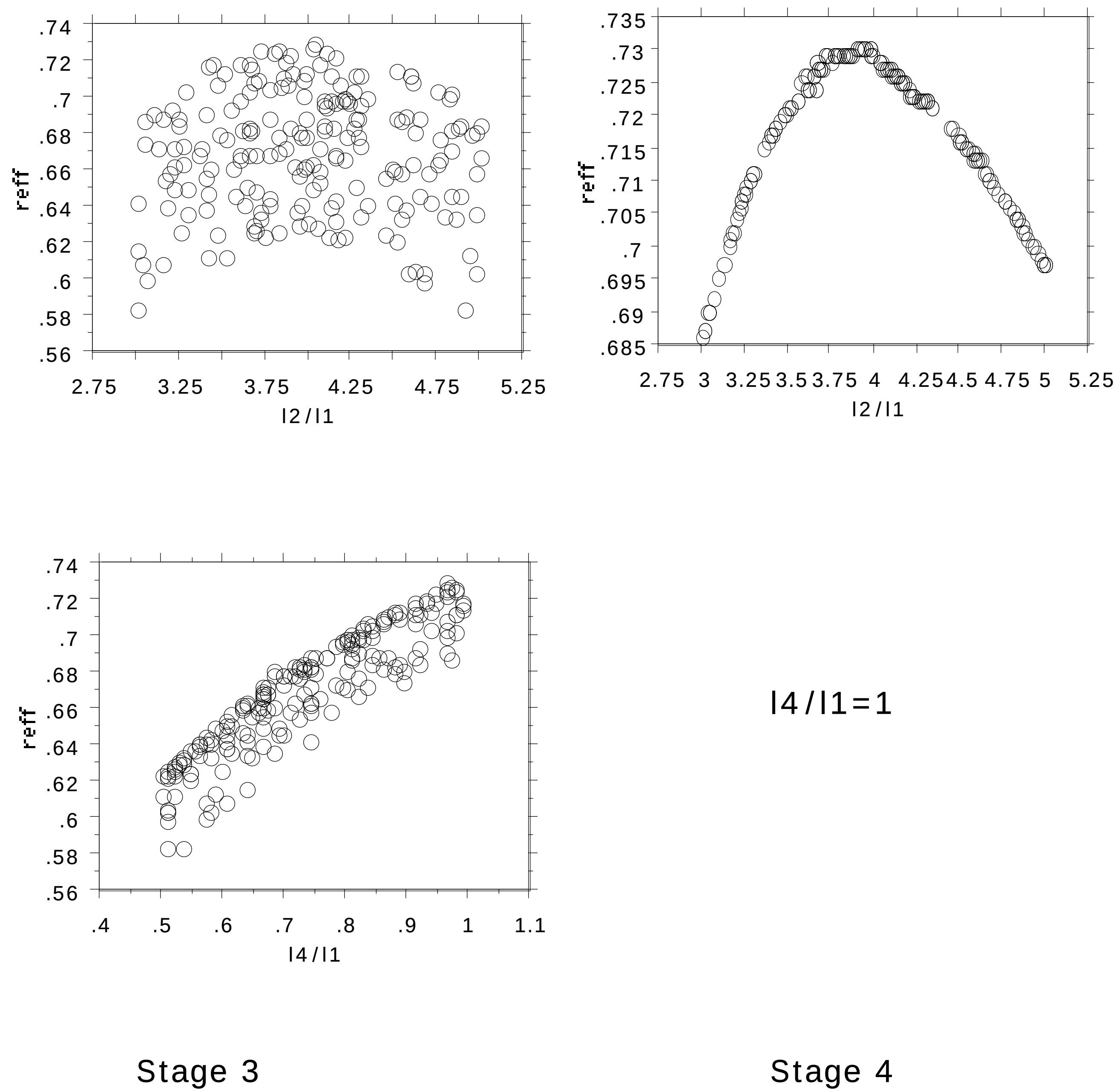


Fig. 21. Stages 3 and 4 of the Monte Carlo exploration. In both of these simulations psiz is constrained to 45° and the sling length/long arm ratio is unity.

Another Monte Carlo run with psiz constrained to be 45° and the sling length=long arm length is shown in Fig. 21 as stage 3. Now the efficiency ranges from 0.58 up to 0.73, so some further gain in the probability that a treb designed within these constraints will have a high efficiency is seen. The general trend of increasing efficiency with increasing l4 is even more apparent than before, but the optimum l2/l1 is not so clearly evident.

Choosing l4 = l1 gives a treb with a counterweight arm that can clear the ground, and gives a further constraint, yielding the Stage 4 results, as also shown in Fig. 21. The range of efficiency in the 200 simulations has been further narrowed to 0.69 to 0.73 and now a clear maximum in the range efficiency in found when l2/l1 = 3.75.

This design process can, of course, be adopted to various other sets of constraints desired, and can be rather quickly carried out. An even faster method, but one that does not reveal much in the way of heuristics, is to set up the problem as for Stage 1, then run a thousand or so simulations, and rank order the results by range efficiency. Successive stages need not be run, and the design parameters are taken from the most efficient design.

Back of the Envelope Design

If you want to start the design of your very own, I would recommend that you play with the equation for range efficiency in this form:

$$R = 2\ \frac{m_1}{m_2}\ h\ \varepsilon_R$$

Put in your CW (m1) and projectile mass (m2) , assume a nominal efficiency of ε_R=0.5. Decide how far you are willing to raise the counterweight (h) by some means, and calculate R. If you like what you see, calculate the length of the short arm of the beam, assuming it is to be placed at a 45° angle with respect to the ground:

$$l_1 = h/1.707.$$

Get the long arm of the beam by multiplying this by four, etc. Choose your counterweight material, and look up it's density. Now draw a design for the counterweight box, and be sure that it is not so long that it hits the ground when the beam is vertical. Determine its volume and assume you fill it with your stuff. Can you get a counterweight mass equal to the m1 assumed above? Iterate this process a couple of times.

When satisfied with the results, refine the design by doing some simulations using either MacTreb or WinTreb simulators. For this, you need the mass of the beam, mb. This is probably the hardest part, and is the most risky. You know the length of the beam required, so to get its mass, you need to estimate its required cross-section to get the volume. To get the cross-section, the best advice, in the absence of any available scientific analysis is to peruse the pictures of previously constructed trebs on the internet, and guess.

Now go build one and have some fun!

An Optimized Design

An alternative design method, is to just use the results given above in the monte-carlo section:

l3/l2=1; l4/l1=1; l2/l1=3.75; l5 = 0.707*l2.

This sets the beam at 45° with respect to the horizontal, and sets all of the proportions. Choosing l1 to be some actual value then sets all of the other lengths. Further, choosing the material of the beam and its cross-section will determine its mass, mb. Choose a projectile that is one-fifth of that, and a counterweight that is 100 times the weight of the projectile. Use a finger angle of 20°.

This process should yield a trebuchet that is about as efficient as one can reasonably expect--about 73 %. Probably little is to be gained by further experimentation with the length parameters--there is not a lot of room between 0.73 and 1.0, after all. It is probably better to spend one's time on construction.

This optimized design, is probably good enough to deserve a name--I call it the "Pretty Darned Good Trebuchet", or the "PDG treb" for short. I know that it works for l1 = 1 ft.

Nonuniform beams

All of the preceding discussion has been based on the assumption that the beam is uniform in cross section. This is generally true in smaller trebs, and is certainly easier to construct. However the beam of the historical trebuchet is generally not uniform—it is usually constructed so that the end nearest the counterweight is very thick, and the end near the sling is much thinner. This has two effects on the motion, both of which are beneficial. The first is that the moment of inertia of the beam is decreased (below that for the uniform beam we have been considering), which lowers the kinetic energy in the beam due to its rotation about the axle. The second advantage is that the center of mass of the beam is closer to the axle, and so requires less energy to raise it to the position where the projectile is released. It is worthwhile to look at these modifications in a little more detail.

The changes in the simulation required for the non-uniform beam are pretty straightforward, and add two more parameters to the model, increasing the complexity of searching for optimum designs somewhat. The required changes have been incorporated into the MacTreb*3.0 and WinTrebStar applications, so this effect can be studied as well.

The center of mass of a uniform beam is, obviously, at a distance (l1+ l2)/2 from the sling end of the beam and the axle is a distance l2 from the same end, so the position of the center of mass, measured from the axle along the beam is rc = (l1+ l2)/2 – l2 = (l1-l2)/2.

The potential energy of a mass mb at a height h above the axle is mb g h, so for the uniform beam the potential energy at a point in the movement when the beam is at an angle θ with respect to the vertical is

$$PEub = mb\ g\ h = mb\ g\ (l1-l2)/2\ \sin(\theta-\pi/2) = -mb\ g\ (l1-l2)/2\ \cos(\theta),$$

which is the expression used previously. For an arbitrarily shaped beam with a center of mass rc from the axle, the potential energy is easily seen to be

$$PEb = -mb\ g\ rc\ \cos(\theta).$$

The position of the center of mass of the beam is relatively simple to measure: balance the beam on a pivot and measure the distance from the pivot to where the axle is going to go. If it is on the long end of the beam, then rc is negative.

The kinetic energy in the beam, since it doesn't translate, but only rotates about the axle, is given by

$$KE_b = \frac{I \dot{\theta}^2}{2}$$

Where I is the moment of inertia of the beam about the axle and theta dot is the rotational velocity about the axle. The moment of inertia of a beam is not as easy to measure as the center of mass, but it is pretty simple to calculate.

Assume that the linear mass density of the beam at a distance x from the axle is s(x). Here x is measured along the beam, and is positive toward the short end of the beam. The linear mass density is the mass per unit length of the beam at x. For a uniform beam, it is given by s(x) = mb/(l1+l2) for x = -l2 up to x=l1. The moment of inertia of a thin beam (neglect its thickness) is given by

$$I = \int_{-l2}^{l1} s(x)\, x^2 dx$$

which, for a uniform thin beam, is then

$$I = \frac{mb}{l1+l2} \left. \frac{x^3}{3} \right|_{-l2}^{l1} = \frac{mb\,(l1^2 - l1\,l2 + l2^2)}{3}$$

A convenient procedure is to compute the moment of inertia using the thin beam approximation, and then use the radius of gyration, rg, for further calculations. The radius of gyration is a length equal to the distance a point mass mb would have to be placed from the axle to have the same moment of inertia. Since the moment of inertia of a point mass m at a distance r from the center of rotation is $m\,r^2$, the radius of gyration of a thin beam about the axle is simply

$$rg = \sqrt{I/mb}$$

which, becomes for a uniform thin beam ,

$$rg = \sqrt{\frac{(l1^2 - l1\,l2 + l2^2)}{3}}$$

For the nominal design, for example, where l1=1 and l2=4, we would have rg = sqrt(13/3) = 2. 08. Trebuchets with a tapered beam would have a smaller radius of gyration than this. Unlike rc, which is zero when the arms of the beam are balanced on the axle, the radius of gyration is always non zero for beams.

For estimation purposes, it is easy to derive simple formulas for beams that are non-uniform to get the radius of gyration. For example, suppose the thickness of the CW end of the beam is k times that of the sling end (both ends still treated as thin parallelpipeds). That is, the beam is shaped like a wedge, as shown in Fig. A1. The origin is taken to be at the axle, the fat end of the beam is l1 long, and the slender end is l2 long.

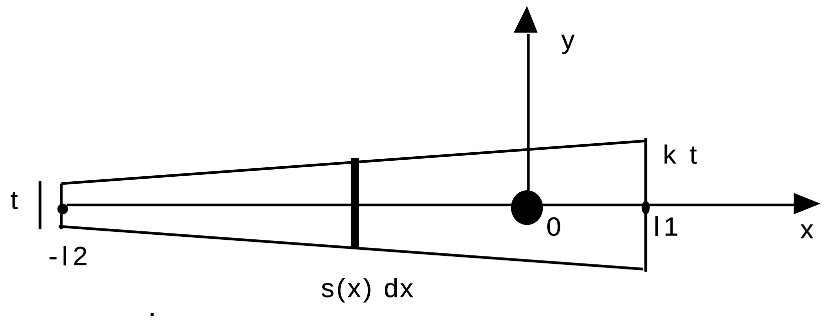


Fig. 22. The linear non-uniform beam with mass per unit length s(x), thickness t at the sling end and k t at the CW end.

Then it is not hard to show that the mass per unit length as a function of x is

$$s(x) = \frac{2\,(l1+k\,l2 + x\,(k-1)}{(1+k)(l1+l2)^2}$$

The radius of gyration of the beam is

$$rg = \sqrt{\frac{(1+3\,k)\,l1^2 - 2\,(1+k)\,l1\,l2 + (3+k)\,l2^2}{6\,(1+k)}}$$

and the center of mass is at

$$x_{cm} = \frac{-2\,l1 + \sqrt{2\,(1+k^2)}\,l1 - 2\,k\,l2 + \sqrt{2\,(1+k^2)}\,l2}{2\,(k-1)}$$

These three equations are seen to reduce to the ones for the uniform beam for k=1. For example, with l1=1, l2 = 4 and k = 4, we get rg = 1.68 and xcm = -0.808. This is to be compared with rg = 2.08 and xcm = -1.5 for the uniform beam.

Materials and strengths needed

All of the forgoing analysis assumes that the beam, axle, CW box, truss, sling and other parts are rigid and strong enough to withstand the forces involved. A discussion of the methods for choosing the materials and dimensions of the the parts constitute a subject fit for further analysis, and a discussion of this is available from the author in the manuscript "Will it Break?". It is a rather simple treatment of the required dimensions of the parts for different designs, and offers some useful rules of thumb and guidlines. See the "Algorithmic Beauty of the Trebuchet" site for details.

Conclusion

A good start on a mechanical analysis of the trebuchet has been described and has shown how it can be applied to design a reasonably efficient trebuchet. Two conclusions are particularly important: the sling length should be equal to the length of the long arm of the beam, and the beam should be set at a 45° angle with respect to the horizontal. The length of the CW arm should be as long as possible, so it just misses hitting the ground (with some margin for error). Use a CW/projectile weight ratio of about 100. You won't go too far wrong!

References

" Trebuchet!", Woodworking, R. Miners, Amer. Woodworking, p 62, Dec. 1997.

""The Trebuchet", P E. Chevedden, L. Eigenbrod, V. Foley, and W. Soedel, Scientific American, p 66, July 1995.

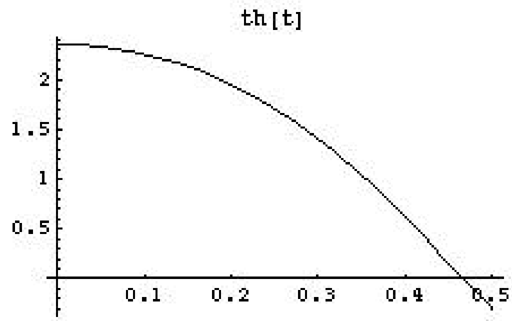
"Building a Better Trebuchet," Leo O'Conner, Mech. Eng. , p 66, Jan. 1994.

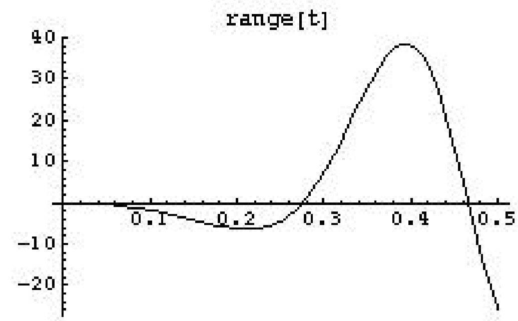
"Algorithmic Beauty of Trebuchets," http://www.members.home.com/dimona.

"Grey Company Web Site", http://www.iinet,au/mine/gctrebs.html.

Appendix 1

```
(*  Solve the seesaw treb problem    *)
(*seesaw1.0.nb DB Siano  Aug. 21, 1997*)
m1=100;m2=1;l1=1;l2=4;g=32;(* the parameters*)

c=g*(l1  m1 - l2  m2)/(m1 l1 l1+ m2 l2 l2);

(*Solve the DE:*)
solss=NDSolve[{th''[t]==-c Sin[th[t]],th[0]==3 Pi/4,th'[0]==0},{th[t]},{t,0,
    1}];
(* get th and the velocity from this solution*)
thint[t_]=Chop[th[t]/.Flatten[solss][[1]]];
v[t_]=l2*thint'[t];(*tangential velocity of the projectile*)

Print["time when th=pi/4 is ",tsolss=t/.FindRoot[thint[t]==Pi/4,{t,.2,.4}]];

Print["vel at th=pi/4 is=",v0pi4=l2*thint'[tsolss]];
Print["range when release is at Pi/4 is  ",r=v0pi4^2/g," ft"];
Print["thoret max range=",rth=N[2 m1 l1 (1+Sin[3 Pi/4])/m2]," ft"];
Print["efficiency=",r/rth];
----
time when th=pi/4 is 0.381169
vel at th=pi/4 is=-34.619
range when release is at Pi/4 is  37.4523 ft
thoret max range = 341.421 ft
efficiency=0.109695
-----
(* Plot the results*)
Plot[{thint[t]},{t,0,.5},PlotLabel->"th[t]"];
(* plot the range as a function of the time at release*)
rang[t_]=2*v[t]*v[t]*Cos[thint[t]]*Sin[thint[t]]/g;
Plot[rang[t],{t,0,.5},PlotLabel->"range[t]"];
```

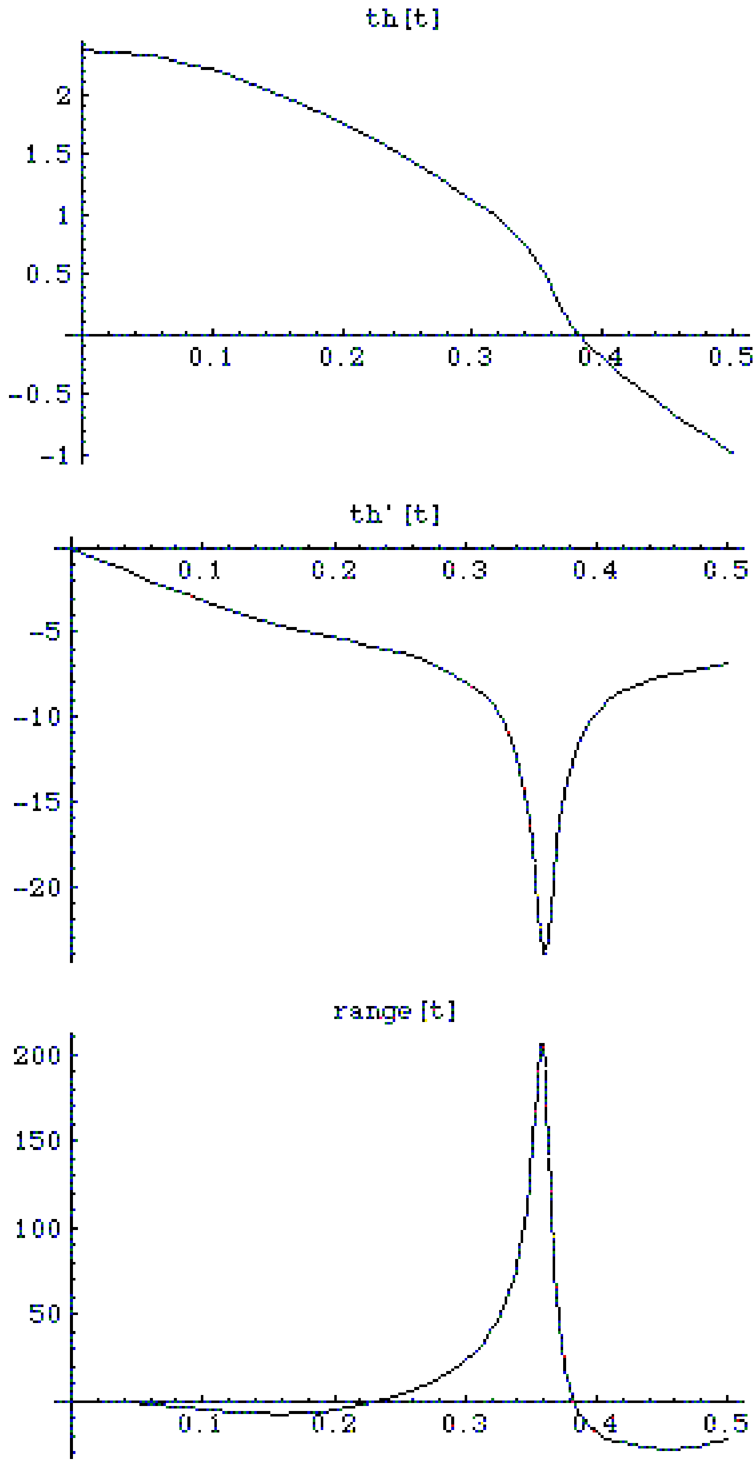

th[t]
2
1.5
1
0.5
0.1
0.2
0.3
0.4
0.5


range[t]
40
30
20
10
-10
-20
0.1
0.2
0.3
0.4
0.5

Appendix II

```
(*  Solve the hinged cw treb problem    *)
(*hingedcw1.0 DB Siano  Aug. 21, 1997*)
th=.;phi=.;psi=.;
l1=.;l2=.;l3=.;l4=.;m1=.;m2=.;m3=.;mb=.;g=.
cn={l1,l2,l3,l4,m1,m2,mb};
x1[th_]:= l1 Sin[th];
y1[th_]:=-l1 Cos[th];
(*coords of long end of  beam*)
x2[th_]:=-l2 Sin[th];
y2[th_]:=l2 Cos[th];
(*coords of short end of counterweight*)
x4[th_,phi_]:=l1 Sin[th]-l4 Sin[phi+th];
y4[th_,phi_]:=-l1 Cos[th]+l4 Cos[phi+th];

(* PE, including the uniform beam of mass mb*)
vt[th_,phi_] :=m1 g y4[th,phi]+m2 g y2[th]-mb g ((l1-l2)/2) Cos[th];
(* KE of the trebuchet*)
ket[th_,phi_]:=
        (m1/2 )*((Dt[x4[th,phi],t,Constants->cn])^2+(
          Dt[y4[th,phi],t,Constants->cn])^2)+
                        (m2/2) *((Dt[x2[th],t,Constants->cn])^2+
(Dt[y2[th],t,Constants->cn])^2 )+
(mb/6) (l2^2-l1 l2 +l1^2) Dt[th,t,Constants->cn]^2;

(* the Lagrangian*)
lagrt[th_,phi_]:=ket[th,phi]-vt[th,phi];
ltrr=lagrt[th,phi]/.{Dt[th,t,Constants\[Rule]{l1,l2,l3,l4,m1,m2,mb}]->thd,
        Dt[phi,t,Constants\[Rule]{l1,l2,l3,l4,m1,m2,mb}]->phid};

(* set up the equations to solve, using the method of Lagrange *)
eqbig=Simplify[{ Dt[D[ltrr,thd],t]-D[ltrr,th]==0,
      Dt[D[ltrr,phid],t]-D[ltrr,phi]==0}/.{Dt[l1,t]->0,Dt[l2,t]->0,
      Dt[l3,t]->0,Dt[l4,t]->0,Dt[mb,t]->0,Dt[m1,t]->0,Dt[m2,t]->0,
      Dt[g,t]->0,
                Dt[th,t]->thd,Dt[phi,t]->phid,Dt[thd,t]->thdd,Dt[phid,t]->phidd}]
---
{(-l1)l4m1((phidd + 2thdd))Cos[phi] +
      l1l4m1phid((phid + 2thd))Sin[phi] +
      1/6((6l4^2m1phidd + 6l1^2m1thdd + 6l4^2m1thdd +
              6l2^2m2thdd + 2l1^2mbthdd - 2l1l2mbthdd +
              2l2^2mbthdd + 6gl1m1Sin[th] -
              6gl2m2Sin[th] + 3gl1mbSin[th] -
              3gl2mbSin[th] - 6gl4m1Sin[phi + th])) == 0,

  l4m1((l4phidd + l4thdd - l1thddCos[phi] - l1thd^2Sin[phi] -
  gSin[phi + th])) == 0}
---

(* the parameters:*)
m1=100;m2=1;mb=0;l1=1;l2=4;l4=1;g=32;
```

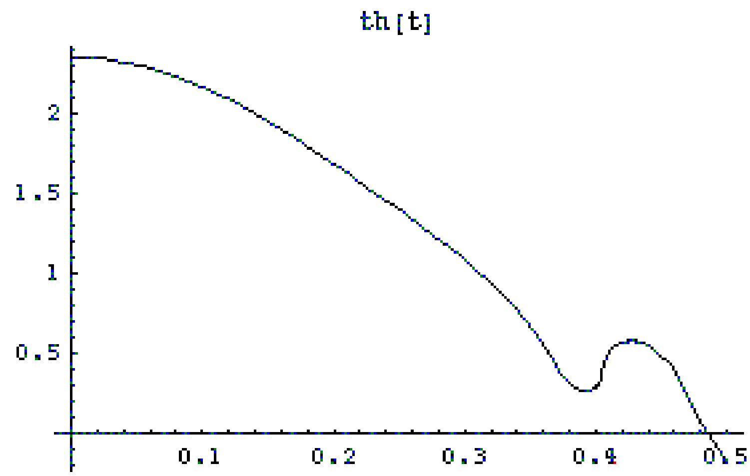
```
l5=l4/Sqrt[2];(*axle height*)

(*the initial conditions:*)
ths=3*Pi/4;(* sets the beam at 45° wrt ground*)
phis=-ths+Pi;(* this makes the cw dangle straight down*)

(* Now put the two equations into a form suitable for NDSolve*)
eqs=eqbig/.{
th->th[t],thd->th'[t],thdd->th''[t],phi->phi[t],phid->phi'[t],phidd->phi''[t]}
----

{100 Sin[phi[t]] phi'[t] (phi'[t] + 2 th'[t]) -
    100 Cos[phi[t]] (phi''[t] + 2 th''[t]) +
    (1 (18432 Sin[th[t]] - 19200 Sin[phi[t] + th[t]] +
         600 phi''[t] + 1296 th''[t])) / 6 == 0,

  100 (-32 Sin[phi[t] + th[t]] - Sin[phi[t]] th'[t] ^2 +
      phi''[t] + th''[t] - Cos[phi[t]] th''[t]) == 0}
-----

(*Solve the DE:*)
solhcw=NDSolve[
    Flatten[{eqs,th[0]==ths,phi[0]==phis,
       th'[0]==0,phi'[0]==0}],{th[t],phi[t]},{t,0.,1}];

(* get th and the velocity from this solution*)
thint[t_]=Chop[th[t]/.Flatten[solhcw][[1]]];
v[t_]=l2*thint'[t];(*tangential velocity of the projectile*)

Print["time when th=pi/4 is ",tsolss=t/.FindRoot[thint[t]==Pi/4,{t,.2,.4}]];

Print["vel at th=pi/4 is=",v0pi4=l2*thint'[tsolss]];
Print["range when release is at Pi/4 is  ",r=v0pi4^2/g," ft"];
Print["thoret max range=",rth=N[2 m1 l1 (1+Sin[3 Pi/4])/m2]," ft"];
Print["efficiency=",r/rth];

-----
time when th=pi/4 is 0.336743
vel at th=pi/4 is=-47.3121
range when release is at Pi/4 is  69.9512 ft
thoret max range=341.421 ft
efficiency=0.204882
-----

(* plot th and its derivative *)
Plot[{thint[t]},{t,0,.5},PlotLabel->"th[t]"];
        gth=Plot[{thint'[t]},{t,0,.5},PlotLabel->"th'[t]"];
(* plot the range as a function of the time at release*)
range[t_]=2*v[t]*v[t]*Cos[thint[t]]*Sin[thint[t]]/g;
Plot[range[t],{t,0,.5},PlotLabel->"range[t]"];
```

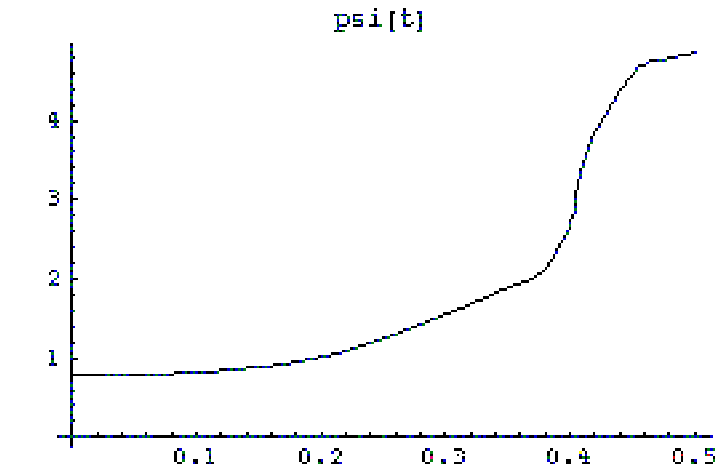

th[t]
2
1.5
1
0.5
-0.5
-1
0.1
0.2
0.3
0.4
0.5
th'[t]
-5
-10
-15
-20
0.1
0.2
0.3
0.4
0.5
range[t]
200
150
100
50
0.1
0.2
0.3
0.4
0.5

Appendix III

```
(*  Solve the sling+cwfree1.0 treb problem    *)
(*seesaw1.0.nb DB Siano  Aug. 21, 1997*)
th =. ; phi =. ; psi =. ; l1 =. ; l2 =. ; l3 =. ; l4 =. ;
  m1 =. ; m2 =. ; m3 =. ; mb =. ; g =.
cn = {l1, l2, l3, l4, m1, m2, mb};
  x1[th_] := l1 Sin[th];
  y1[th_] := -l1 Cos[th]; x2[th_] := -l2 Sin[th];
  y2[th_] := l2 Cos[th];
  x4[th_, phi_] := l1 Sin[th] - l4 Sin[phi + th];
  y4[th_, phi_] := -l1 Cos[th] + l4 Cos[phi + th];
  x3[th_, psi_] := -(l3 Sin[psi - th] + l2 Sin[th]);
  y3[th_, psi_] := -(l3 Cos[psi - th] - l2 Cos[th]);
  vt[th_, phi_, psi_] :=
                                                     l1 - l2
   m1 g y4[th, phi] + m2 g y3[th, psi] - mb g ------- Cos[th]
                                                        2

ket[th_, phi_, psi_] :=

   m1                                      2
   -- (Dt[x4[th, phi], t, Constants -> cn]  +
   2
                                           2
       Dt[y4[th, phi], t, Constants -> cn] ) +

    m2                                      2
    -- (Dt[x3[th, psi], t, Constants -> cn]  +
    2
                                           2
       Dt[y3[th, psi], t, Constants -> cn] ) +

    mb    2             2                            2
    -- (l2  - l1 l2 + l1 ) Dt[th, t, Constants -> cn] ;
    6

  lagrt[th_, phi_] := ket[th, phi, psi] - vt[th, phi, psi];
  ltrr = lagrt[th, phi] /.
    {Dt[th, t, Constants -> {l1, l2, l3, l4, m1, m2, mb}] ->
     thd, Dt[phi, t, Constants ->
       {l1, l2, l3, l4, m1, m2, mb}] -> phid,
    Dt[psi, t, Constants ->
       {l1, l2, l3, l4, m1, m2, mb}] -> psid};


eqbig = Simplify[{Dt[D[ltrr, thd], t] - D[ltrr, th] == 0,
    Dt[D[ltrr, phid], t] - D[ltrr, phi] == 0,
    Dt[D[ltrr, psid], t] - D[ltrr, psi] == 0} /.
   {Dt[l1, t] -> 0, Dt[l2, t] -> 0, Dt[l3, t] -> 0,
    Dt[l4, t] -> 0, Dt[mb, t] -> 0, Dt[m1, t] -> 0,
    Dt[m2, t] -> 0, Dt[g, t] -> 0, Dt[th, t] -> thd,
    Dt[phi, t] -> phid, Dt[psi, t] -> psid,
```

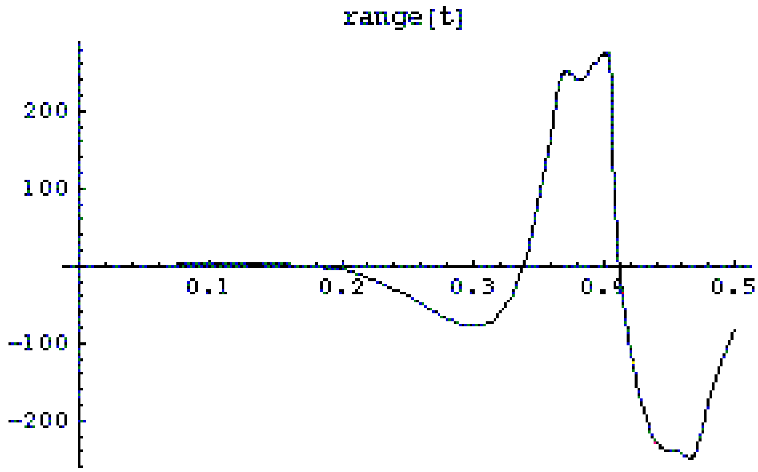
```
    Dt[thd, t] -> thdd, Dt[phid, t] -> phidd,
    Dt[psid, t] -> psidd}];

------------
 l4**2*m1*phidd - l3**2*m2*psidd +
    -   l1**2*m1*thdd + l4**2*m1*thdd + l2**2*m2*thdd +
    -   l3**2*m2*thdd + (l1**2*mb*thdd)/3. -
    -   (l1*l2*mb*thdd)/3. + (l2**2*mb*thdd)/3. -
    -   l1*l4*m1*(phidd + 2*thdd)*Cos(phi) +
    -   l2*l3*m2*(psidd - 2*thdd)*Cos(psi) +
    -   l1*l4*m1*phid**2*Sin(phi) +
    -   2*l1*l4*m1*phid*thd*Sin(phi) -
    -   l2*l3*m2*psid**2*Sin(psi) +
    -   2*l2*l3*m2*psid*thd*Sin(psi) -
    -   g*l3*m2*Sin(psi - th) + g*l1*m1*Sin(th) -
    -   g*l2*m2*Sin(th) + (g*l1*mb*Sin(th))/2. -
    -   (g*l2*mb*Sin(th))/2. - g*l4*m1*Sin(phi + th).eq.0,

l4*m1*(l4*phidd + l4*thdd - l1*thdd*Cos(phi) -
    -     l1*thd**2*Sin(phi) - g*Sin(phi + th)).eq.0,

 l3*m2*(l3*psidd - l3*thdd + l2*thdd*Cos(psi) -
    -     l2*thd**2*Sin(psi) + g*Sin(psi - th)).eq.0

-----------------
(* input the parameters here*)
m1 = 100; m2 = 1; mb = 0; l1 = 1; l2 = 4;
l3 = 3.25; l4 = 1; g = 32; l5 = l4/Sqrt[2];
(psis = ths - Pi/2; ); ths = (3*Pi)/4; phis = -ths + Pi;
(*set up the equations *)
eqs = eqbig /.
   {th -> th[t], thd -> Derivative[1][th][t],
    thdd -> Derivative[1][Derivative[1][th]][t],
    phi -> phi[t], phid -> Derivative[1][phi][t],
    phidd -> Derivative[1][Derivative[1][phi]][t],
    psi -> psi[t], psid -> Derivative[1][psi][t],
    psidd -> Derivative[1][Derivative[1][psi]][t]};

(*Solve the DE:*)
solslcw=NDSolve[
      Flatten[{eqs,th[0]==ths,phi[0]==phis,psi[0]==psis,
          th'[0]==0,phi'[0]==0,psi'[0]==0}],{th[t],phi[t],psi[t]},{t,0.,1}];


(* get th, psi and the velocity from this solution*)
thint[t_]=Chop[th[t]/.Flatten[solslcw][[1]]];
psiint[t_]=Chop[psi[t]/.Flatten[solslcw][[3]]];
Plot[thint[t],{t,0,.5},PlotLabel->"th[t]"];
Plot[psiint[t],{t,0,.5},PlotLabel->"psi[t]"];
```





(* now to get the range, we first need to get the velocity of the projectile*)

```
vx3[t_]:=Chop[-l3 Cos[psiint[t]-thint[t]]*(psiint'[t]-thint'[t])-
    l2*Cos[thint[t]]*thint'[t]];
vy3[t_]:=Chop[
   l3*(psiint'[t]-thint'[t])*Sin[psiint[t]-thint[t]]-
    l2*thint'[t]*Sin[thint[t]]];
range[t_]=2 vx3[t]* vy3[t]/g;
Plot[range[t],{t,0,.5},PlotLabel->"range[t]"]
```




```
(* get the maximum range *)
tstrt=0;tend=.5;del=.001;
rangetime=Table[{range[t],t},{t,tstrt,tend,del*(tend-tstrt)}];
allowedranges=Transpose[rangetime];
mallowedrange=Max[First[allowedranges]];
postallowedrange=First[Flatten[Position[rangetime,mallowedrange]]];
tallowedrange=Last[rangetime[[postallowedrange]]];
Print[" The maximum range is ",mallowedrange,"  at time ",
  tallowedrange, " s"];
Print["theta at time of release is ",180/Pi*thint[tallowedrange]," deg"];
        Print["psi at time of release is ",180/Pi*psiint[tallowedrange]," deg"];
Print["thoret max range=",rth=N[2 m1 l1 (1+Sin[Pi/4])/m2]," ft"];
Print["efficiency=",mallowedrange/rth];
------
The maximum range is 275.793  at time 0.4045 s

theta at time of release is 20.7797 deg

psi at time of release is 169.201 deg

thoret max range=341.421 ft

efficiency=0.80778
```

Appendix IV

Equations for the second derivatives of the angles for the constrained portion of the throw are shown here. Subsidiary variables that are used to speed the calculation are defined first. The notation is to use "d" to mean first derivative and "dd" to mean the second derivative; sq means second power, cu the third, qt the fourth, and qn the fifth power.

```
sth=sin(th)
cth=cos(th)
sphi=sin(phi)
cphi=cos(phi)
```

```
thdsq=thd*thd
phidsq=phid*phid
sthsq=sth*sth
sthcu=sthsq*sth
cthsq=cth*cth
cthcu=cthsq*cth

cphisq=cphi*cphi

rad= Sqrt(1 - (l2sq*(cth + spsis)*(cth + spsis))/l3sq)
r2=rad*rad
r3=r2*rad
r4=r2*r2

// for th in the constrained portion of the throw:

c1=   -6*l2cu*l3sq*m2*r3*thdsq*cthcu + 6*l1*l3cu*l4*m1*phidsq*r4*sphi
c2=  12*l1*l3cu*l4*m1*phid*r4*thd*sphi + 6*l1*l3cu*l4*m1*r4*thdsq*sphi
c3=  -6*l1sq*l3cu*m1*r4*thdsq*cphi*sphi
c4= -6*l2cu*l3sq*m2*r3*thdsq*cthsq*spsis + 6*g*l1*l3cu*m1*r4*sth
c5=  3*g*l1*l3cu*mb*r4*sth - 3*g*l2*l3cu*mb*r4*sth
c6=  6*l2sq*l3cu*m2*r2*thdsq*cth*sth
c7=  -12*l2sq*l3cu*m2*r4*thdsq*cth*sth
c8=  12*l2cu*l3sq*m2*r3*thdsq*cth*sthsq
c9=  6*l2qn*m2*rad*thdsq*cthcu*sthsq
c10=  6*l2cu*l3sq*m2*r3*thdsq*spsis*sthsq
c11=  12*l2qn*m2*rad*thdsq*cthsq*spsis*sthsq
c12=  6*l2qn*m2*rad*thdsq*cth*spsissq*sthsq
c13=  -6*l2qt*l3*m2*thdsq*cth*sthcu - 6*l2qt*l3*m2*thdsq*spsis*sthcu
c14=  -6*g*l1*l3cu*m1*r4*cphi*Sin(phi + th)
cnum=c1+c2+c3+c4+c5+c6+c7+c8+c9+c10+c11+c12+c13+c14

d1= -6*l1sq*l3cu*m1*r4 - 6*l2sq*l3cu*m2*r4 - 2*l1sq*l3cu*mb*r4
d2=  2*l1*l2*l3cu*mb*r4 - 2*l2sq*l3cu*mb*r4 + 6*l1sq*l3cu*m1*r4*cphisq
d3=  12*l2cu*l3sq*m2*r3*cthsq*sth
d4=  12*l2cu*l3sq*m2*r3*cth*spsis*sth - 6*l2sq*l3cu*m2*r2*sthsq
d5=  12*l2sq*l3cu*m2*r4*sthsq
ddenom=d1+d2+d3+d4+d5

//return the second derivative of th.
thddcret=cnum/ddenom
//-----------------------------------------------------------------------------------------------------------
```

The equations for phi in the constrained portion of the throw:

```
a(1)=-(l1*l4cu*m1sq*phidsq*sphi) + cphi*l1sq*l4sq*m1sq*phidsq*sphi +
cphi*g*l1sq*l4*m1sq*sth
     a(2)=   -g*l1*l4sq*m1sq*sth + (cphi*g*l1sq*l4*m1*mb*sth)/2. -
(cphi*g*l1*l2*l4*m1*mb*sth)/2.
     a(3)=    -(g*l1*l4sq*m1*mb*sth)/2. + (g*l2*l4sq*m1*mb*sth)/2. -
2*l1*l4cu*m1sq*phid*sphi*thd
```

```
a(4)=    2*cphi*l1sq*l4sq*m1sq*phid*sphi*thd -
(cphi*cthcu*l1*l2cu*l4*m1*m2*thdsq)/(l3*rad)
    a(5)=   (cthcu*l2cu*l4sq*m1*m2*thdsq)/(l3*rad) - l1cu*l4*m1sq*sphi*thdsq
    a(6)=   -l1*l4cu*m1sq*sphi*thdsq + 2*cphi*l1sq*l4sq*m1sq*sphi*thdsq -
l1*l2sq*l4*m1*m2*sphi*thdsq
    a(7)=   -(l1cu*l4*m1*mb*sphi*thdsq)/3. + (l1sq*l2*l4*m1*mb*sphi*thdsq)/3.
    a(8)=   -(l1*l2sq*l4*m1*mb*sphi*thdsq)/3. -
(cphi*cthsq*l1*l2cu*l4*m1*m2*spsis*thdsq)/(l3*rad)
    a(9)=    (cthsq*l2cu*l4sq*m1*m2*spsis*thdsq)/(l3*rad) -
2*cphi*cth*l1*l2sq*l4*m1*m2*sth*thdsq
    a(10)=    2*cth*l2sq*l4sq*m1*m2*sth*thdsq +
(cphi*cth*l1*l2sq*l4*m1*m2*sth*thdsq)/r2
    a(11)=    -(cth*l2sq*l4sq*m1*m2*sth*thdsq)/r2
    a(12)=    (2*cthsq*l1*l2cu*l4*m1*m2*sphi*sth*thdsq)/(l3*rad)
    a(13)=    (2*cth*l1*l2cu*l4*m1*m2*sphi*spsis*sth*thdsq)/(l3*rad)
    a(14)=    -(cphi*cth*l1*l2qt*l4*m1*m2*sthcu*thdsq)/(l3sq*r4)
    a(15)=   (cth*l2qt*l4sq*m1*m2*sthcu*thdsq)/(l3sq*r4)
    a(16)=    -(cphi*l1*l2qt*l4*m1*m2*spsis*sthcu*thdsq)/(l3sq*r4)
    a(17)=    (l2qt*l4sq*m1*m2*spsis*sthcu*thdsq)/(l3sq*r4)
    a(18)=    (cphi*cthcu*l1*l2qn*l4*m1*m2*sthsq*thdsq)/(l3cu*r3)
    a(19)=    -(cthcu*l2qn*l4sq*m1*m2*sthsq*thdsq)/(l3cu*r3)
    a(20)=    (2*cphi*cth*l1*l2cu*l4*m1*m2*sthsq*thdsq)/(l3*rad)
    a(21)=    -(2*cth*l2cu*l4sq*m1*m2*sthsq*thdsq)/(l3*rad) +
2*l1*l2sq*l4*m1*m2*sphi*sthsq*thdsq
    a(22)=    -(l1*l2sq*l4*m1*m2*sphi*sthsq*thdsq)/r2
    a(23)=   (2*cphi*cthsq*l1*l2qn*l4*m1*m2*spsis*sthsq*thdsq)/(l3cu*r3)
    a(24)=   -(2*cthsq*l2qn*l4sq*m1*m2*spsis*sthsq*thdsq)/(l3cu*r3)
    a(25)=   (cphi*l1*l2cu*l4*m1*m2*spsis*sthsq*thdsq)/(l3*rad)
    a(26)=   -(l2cu*l4sq*m1*m2*spsis*sthsq*thdsq)/(l3*rad)
    a(27)=    (cphi*cth*l1*l2qn*l4*m1*m2*spsissq*sthsq*thdsq)/(l3cu*r3)
    a(28)=    -(cth*l2qn*l4sq*m1*m2*spsissq*sthsq*thdsq)/(l3cu*r3) -
g*l1sq*l4*m1sq*Sin(phi + th)
    a(29)=    cphi*g*l1*l4sq*m1sq*Sin(phi + th) - g*l2sq*l4*m1*m2*Sin(phi + th)
    a(30)=    -(g*l1sq*l4*m1*mb*Sin(phi + th))/3. + (g*l1*l2*l4*m1*mb*Sin(phi +
th))/3.
    a(31)=    -(g*l2sq*l4*m1*mb*Sin(phi + th))/3.
    a(32)=    (2*cthsq*g*l2cu*l4*m1*m2*sth*Sin(phi + th))/(l3*rad)
    a(33)=   (2*cth*g*l2cu*l4*m1*m2*spsis*sth*Sin(phi + th))/(l3*rad)
    a(34)=    2*g*l2sq*l4*m1*m2*sthsq*Sin(phi + th) -
(g*l2sq*l4*m1*m2*sthsq*Sin(phi + th))/r2


    b(1)=  -(l1sq*l4sq*m1sq) + cphisq*l1sq*l4sq*m1sq - l2sq*l4sq*m1*m2 -
(l1sq*l4sq*m1*mb)/3.
    b(2)=    (l1*l2*l4sq*m1*mb)/3. - (l2sq*l4sq*m1*mb)/3. +
(2*cthsq*l2cu*l4sq*m1*m2*sth)/(l3*rad)
    b(3)=   (2*cth*l2cu*l4sq*m1*m2*spsis*sth)/(l3*rad) + 2*l2sq*l4sq*m1*m2*sthsq
    b(4)=    -(l2sq*l4sq*m1*m2*sthsq)/r2
phiddtopc=0.
nitems=34
for i=1 to 34
phiddtopc=phiddtopc+a(i)
next
```

```
phiddbotc=b(1)+b(2)+b(3)+b(4)

//return the second derivative of phi:
phiddcres=phiddtopc/phiddbotc
//----------------------------------------------------------------------------------------------------------
```

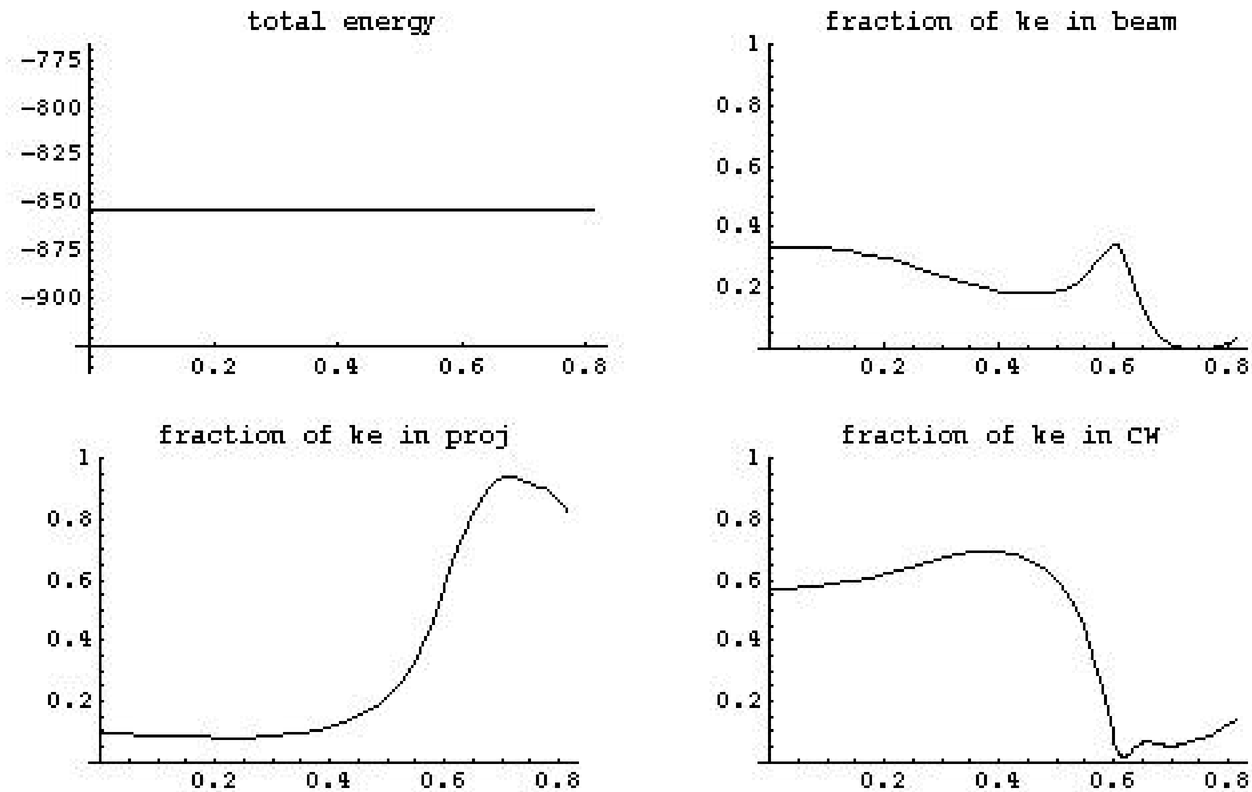
The three equations for the second dervatives of the angles during the unconstrained throw are shown here.

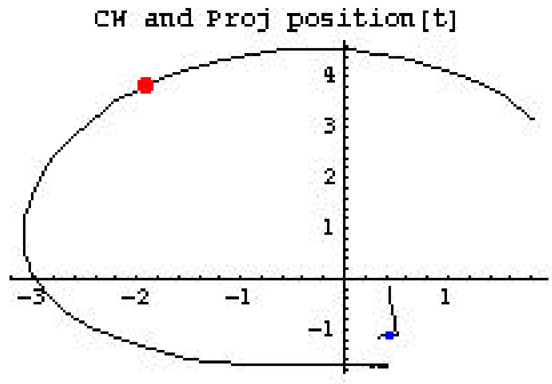
```
//thdd
 a1=6*l1*l4*m1*phid*phid*sphi + 12*l1*l4*m1*phid*thd*sphi
  a2=6*l1*l4*m1*thd*thd*sphi - 6*l1sq*m1*thd*thd*cphi*sphi
  a3=-6*l2*l3*m2*psid*psid*spsi + 12*l2*l3*m2*psid*thd*spsi
  a4=-6*l2*l3*m2*thd*thd*spsi + 6*l2sq*m2*thd*thd*cpsi*spsi
  a5=-6*g*l2*m2*cpsi*(spsi*cth-sth*cpsi) + 6*g*l1*m1*sth - 6*g*l2*m2*sth
  a6=3*g*l1*mb*sth - 3*g*l2*mb*sth - 6*g*l1*m1*cphi*(sphi*cth+sth*cphi)
  thtop=a1+a2+a3+a4+a5+a6
  b1=-6*l1sq*m1 - 6*l2sq*m2 - 2*l1sq*mb + 2*l1*l2*mb - 2*l2sq*mb
  b2=6*l1sq*m1*cphi*cphi + 6*l2sq*m2*cpsi*cpsi
  thbot=b1+b2
  thdd=thtop/thbot

//phidd
   c1=-6*l1*l4sq*m1*phid*phid*sphi - 12*l1*l4sq*m1*phid*thd*sphi -
6*l1cu*m1*thd*thd*sphi
    c2=-6*l1*l4sq*m1*thd*thd*sphi - 6*l1*l2sq*m2*thd*thd*sphi -
2*l1cu*mb*thd*thd*sphi
    c3=2*l1sq*l2*mb*thd*thd*sphi - 2*l1*l2sq*mb*thd*thd*sphi +
6*l1sq*l4*m1*phid*phid*cphi*sphi
    c4=12*l1sq*l4*m1*phid*thd*cphi*sphi + 12*l1sq*l4*m1*thd*thd*cphi*sphi
    c5=6*l1*l2sq*m2*thd*thd*cpsi*cpsi*sphi + 6*l2*l3*l4*m2*psid*psid*spsi
    c6=-12*l2*l3*l4*m2*psid*thd*spsi + 6*l2*l3*l4*m2*thd*thd*spsi
    c7=-6*l1*l2*l3*m2*psid*psid*cphi*spsi + 12*l1*l2*l3*m2*psid*thd*cphi*spsi
    c8=-6*l1*l2*l3*m2*thd*thd*cphi*spsi - 6*l2sq*l4*m2*thd*thd*cpsi*spsi
    c9=6*l1*l2sq*m2*thd*thd*cphi*cpsi*spsi + 6*g*l2*l4*m2*cpsi*(spsi*cth-sth*cpsi)
    c10=-6*g*l1*l2*m2*cphi*cpsi*(spsi*cth-sth*cpsi) - 6*g*l1*l4*m1*sth +
6*g*l2*l4*m2*sth
    c11=-3*g*l1*l4*mb*sth + 3*g*l2*l4*mb*sth + 6*g*l1sq*m1*cphi*sth
    c12=-6*g*l1*l2*m2*cphi*sth + 3*g*l1sq*mb*cphi*sth - 3*g*l1*l2*mb*cphi*sth
    c13=-6*g*l1sq*m1*(sphi*cth+sth*cphi) - 6*g*l2sq*m2*(sphi*cth+sth*cphi) -
2*g*l1sq*mb*(sphi*cth+sth*cphi)
    c14=2*g*l1*l2*mb*(sphi*cth+sth*cphi) - 2*g*l2sq*mb*(sphi*cth+sth*cphi) +
6*g*l1*l4*m1*cphi*(sphi*cth+sth*cphi)
    c15=6*g*l2sq*m2*cpsi*cpsi*(sphi*cth+sth*cphi)


    d1=-6*l1sq*l4*m1 - 6*l2sq*l4*m2 - 2*l1sq*l4*mb + 2*l1*l2*l4*mb - 2*l2sq*l4*mb
    d2=6*l1sq*l4*m1*cphi*cphi + 6*l2sq*l4*m2*cpsi*cpsi

    phitop=c1+c2+c3+c4+c5+c6+c7+c8+c9+c10+c11+c12+c13+c14+c15
    phibot=d1+d2
   phidd=phitop/phibot
//----------------------------------------------------------------------------------------------------------
```

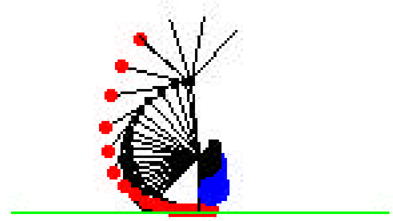
```
//psidd

e1=l1*l3sq*l4cu*m1sq*m2*phid*phid*sphi + 2*l1*l3sq*l4cu*m1sq*m2*phid*thd*sphi
  e2=l1*l3sq*l4cu*m1sq*m2*thd*thd*sphi -
l1sq*l3sq*l4sq*m1sq*m2*thd*thd*cphi*sphi
  e3=-l1*l2*l3*l4cu*m1sq*m2*phid*phid*cpsi*sphi -
2*l1*l2*l3*l4cu*m1sq*m2*phid*thd*cpsi*sphi
  e4=-l1*l2*l3*l4cu*m1sq*m2*thd*thd*cpsi*sphi
  e5=l1sq*l2*l3*l4sq*m1sq*m2*thd*thd*cphi*cpsi*sphi -
l2*l3cu*l4sq*m1*m2sq*psid*psid*spsi
  e6=2*l2*l3cu*l4sq*m1*m2sq*psid*thd*spsi - l1sq*l2*l3*l4sq*m1sq*m2*thd*thd*spsi
  e7=-l2cu*l3*l4sq*m1*m2sq*thd*thd*spsi - l2*l3cu*l4sq*m1*m2sq*thd*thd*spsi
  e8=-(l1sq*l2*l3*l4sq*m1*m2*mb*thd*thd*spsi)/3. +
(l1*l2sq*l3*l4sq*m1*m2*mb*thd*thd*spsi)/3.
  e9=-(l2cu*l3*l4sq*m1*m2*mb*thd*thd*spsi)/3. +
l1sq*l2*l3*l4sq*m1sq*m2*thd*thd*cphi*cphi*spsi
  e10=l2sq*l3sq*l4sq*m1*m2sq*psid*psid*cpsi*spsi
  e11=-2*l2sq*l3sq*l4sq*m1*m2sq*psid*thd*cpsi*spsi
  e12=2*l2sq*l3sq*l4sq*m1*m2sq*thd*thd*cpsi*spsi +
g*l1sq*l3*l4sq*m1sq*m2*(spsi*cth-sth*cpsi)
  e13=g*l2sq*l3*l4sq*m1*m2sq*(spsi*cth-sth*cpsi) +
(g*l1sq*l3*l4sq*m1*m2*mb*(spsi*cth-sth*cpsi))/3.
  e14=-(g*l1*l2*l3*l4sq*m1*m2*mb*(spsi*cth-sth*cpsi))/3. +
(g*l2sq*l3*l4sq*m1*m2*mb*(spsi*cth-sth*cpsi))/3.
  e15=-g*l1sq*l3*l4sq*m1sq*m2*cphi*cphi*(spsi*cth-sth*cpsi) -
g*l2*l3sq*l4sq*m1*m2sq*cpsi*(spsi*cth-sth*cpsi)
  e16=g*l1*l3sq*l4sq*m1sq*m2*sth - g*l2*l3sq*l4sq*m1*m2sq*sth
  e17=(g*l1*l3sq*l4sq*m1*m2*mb*sth)/2. - (g*l2*l3sq*l4sq*m1*m2*mb*sth)/2.
  e18=-g*l1*l2*l3*l4sq*m1sq*m2*cpsi*sth + g*l2sq*l3*l4sq*m1*m2sq*cpsi*sth
  e19=-(g*l1*l2*l3*l4sq*m1*m2*mb*cpsi*sth)/2. +
(g*l2sq*l3*l4sq*m1*m2*mb*cpsi*sth)/2.
  e20=-g*l1*l3sq*l4sq*m1sq*m2*cphi*(sphi*cth+sth*cphi) +
g*l1*l2*l3*l4sq*m1sq*m2*cphi*cpsi*(sphi*cth+sth*cphi)

psitop=e1+e2+e3+e4+e5+e6+e7+e8+e9+e10+e11+e12+e13+e14+e15+e16+e17+e18+e19
+e20

  f1=-(l1sq*l3sq*l4sq*m1sq*m2) - l2sq*l3sq*l4sq*m1*m2sq -
(l1sq*l3sq*l4sq*m1*m2*mb)/3.
  f2=(l1*l2*l3sq*l4sq*m1*m2*mb)/3. - (l2sq*l3sq*l4sq*m1*m2*mb)/3.
  f3=l1sq*l3sq*l4sq*m1sq*m2*cphi*cphi
  f4=l2sq*l3sq*l4sq*m1*m2sq*cpsi*cpsi
  psibot=f1+f2+f3+f4
  psidd=psitop/psibot
```

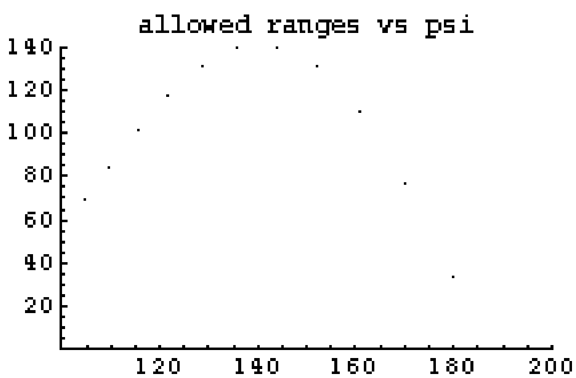
Appendix V

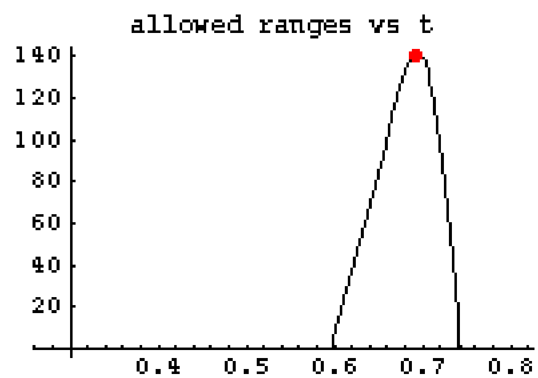
This shows the results from a Mathematica simulation for a "nominal" trebuchet in which, l1=1, l2=4, l3=3.5, l4=1, l5=l2/sqrt(2), m1=100, m2=1, mb=5:

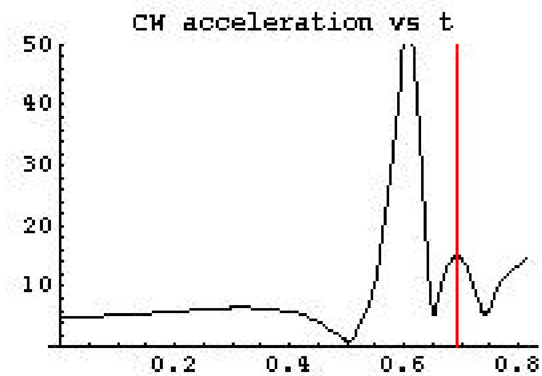


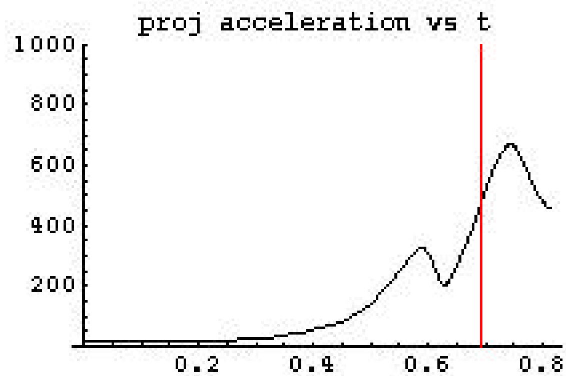

CM and Proj position[t]
4
3
2
1
-3
-2
-1
1
-1

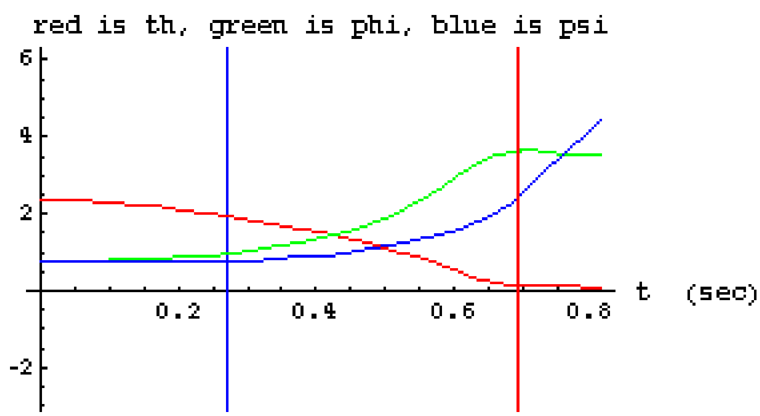



allowed ranges vs psi
140
120
100
80
60
40
20
120
140
160
180
200


allowed ranges vs t
140
120
100
80
60
40
20
0.4
0.5
0.6
0.7
0.8


CW acceleration vs t
50
40
30
20
10
0.2
0.4
0.6
0.8


proj acceleration vs t
1000
800
600
400
200
0.2
0.4
0.6
0.8


red is th, green is phi, blue is psi
6
4
2
-2
0.2
0.4
0.6
0.8
t (sec)

```
nominal treb design
 lshrt= 0.6096 m  llong=2.4384 m lsling= 2.1336 m lcw= 0.6096 m laxle= 1.72421 m
l1=0.6096m   l2/l1=4.  l3/l1= 3.5   l4/l1= 1.   l5/l1= 2.82843
 mcw= 363.636 kg   mproj= 3.63636 kg      mbeam=24.5455 kg
m2= 3.63636 kg     m1/m2=100.     mb/m2=6.75
 The starting angles (th, phi, psi) were  135.  45.  45. degrees
 The maximum range allowed is 140.693 m
 The black box range is 208.13 m
 The distance the CW fell is  1.04065 m
 The range efficiency is 0.675984
 It leaves at an angle wrt the horizontal at 43.8276 degrees  at t= 0.693559 s
 with a velocity = 37.1476 m/s and an energy = 2508.99 J.
 It leaves the slide at time =0.270922
 The available potential energy in the cw was 3708.51 J.
 When the projectile leaves, the angles are {7.844, 208.4, 140.5} degrees
 The fraction of the cw energy deposited in the projectile was =  0.68
 Total kinetic energy =2694.38 J.
The acceleration of the projectile at an instant   just before release is  474.432 m s^-2
 The rate of change of psi at the same moment is 14.5774 rad/s
 The rate of change of th  at the same moment is -1.48823 rad/s
 Set the finger angle (steel on steel) to 20.4505 degrees.
```