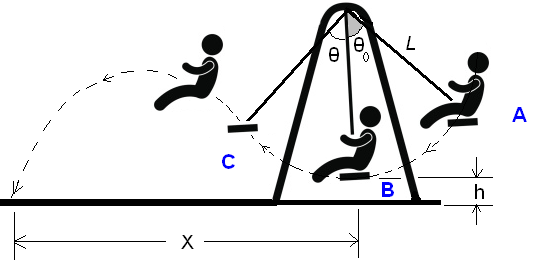
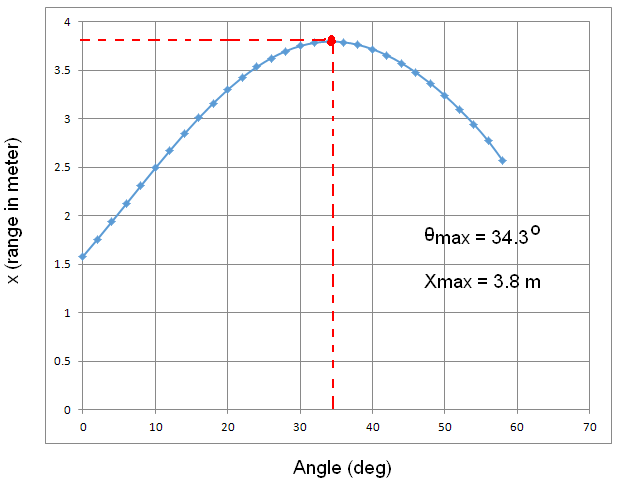
**EP208 Additional Document**

*Getting the greatest distance away from the swing*

As shown in figure, a child starts to swing at an initial angle θ0 = 60o from point A. Then, he passes though the minimum point B. At point C where the angular position is θ < θ0 he jumps from swing and falls down at a distance x from point B. Determine the optimal value of θ such that he can reach the maximum distance from the minimum point of the swing. Assume that the height and length of the swing are *h* = 0.5 m and *L* = 2.5 m respectively. The initial angle is θ0 = 60o.



|  |  |
| --- | --- |
| Ignoring the air resistance Physics says: | Output Data |
| Input θ  L = 2.5;  H = 0.5;  θ0 = 60\*M\_PI/180;  g = 9.8;  v = sqrt(2\*g\*L\*(cos(θ)-cos(θ0)));  a = -g/(2\*v\*v\*cos(θ)\*cos(θ));  b = tan(θ);  c = L\*(1-cos(θ))+h;  x1= L\*sin(θ);  x2= (-b-sqrt(b\*b-4\*a\*c))/(2\*a);  Range = x1 + x2; | **Θ (deg) Range(meter)**  **------ -----------**  0 1.58114  2 1.75835  4 1.93994  6 2.12418  8 2.30903  10 2.49224  12 2.67143  14 2.84419  16 3.00818  18 3.16117  20 3.30114  22 3.42626  24 3.53498  26 3.62602  28 3.69836  30 3.75129  32 3.78437  34 3.79745  36 3.79063  38 3.76427  40 3.71895  42 3.65545  44 3.57469  46 3.47769  48 3.36550  50 3.23899  52 3.09862  54 2.94384  56 2.77136  58 2.56849 |



C++ code:

#include <iostream>

#include <cmath>

using namespace std;

// function to minimize (the Range)

double f(double theta){

double L = 2.5, h=0.5, theta0=60\*M\_PI/180, g=9.8;

double v = sqrt(2\*g\*L\*(cos(theta)-cos(theta0)));

double x1= L\*sin(theta);

double a =-g/(2\*v\*v\*cos(theta)\*cos(theta));

double b = tan(theta);

double c = L\*(1-cos(theta))+h;

double x2= (-b-sqrt(b\*b-4\*a\*c))/(2\*a);

double R = x1+x2;

return R;

}

// this function return the optimum angle

double optimize(double x0=0.5, double tol=1.0e-6, double h=5.0e-4){

double x = x0, err;

int iter = 1;

do{

err = 0.5\*h\*(f(x+h)-f(x-h))/(f(x+h)-2\*f(x)+f(x-h));

x = x - err;

iter++;

}while(fabs(err)>tol || iter>50);

if(iter>50) cout << "Algorithm does not converge since iter > 50.\n";

return x;

}

// the main program prints the optimum angle and range

int main(){

double t = optimize(0.6);

cout << t\*180/M\_PI << " --> " << f(t) << endl;

}