**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 θL = 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 1.581142 1.758354 1.939946 2.124188 2.3090310 2.4922412 2.6714314 2.8441916 3.0081818 3.1611720 3.3011422 3.4262624 3.5349826 3.6260228 3.6983630 3.7512932 3.7843734 3.7974536 3.7906338 3.7642740 3.7189542 3.6554544 3.5746946 3.4776948 3.3655050 3.2389952 3.0986254 2.9438456 2.7713658 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;

}