# define DTOR (3.141592654/180.0)
main(argc, argv)
    int argc;
    char *argv[];
    
    int i, j, k;
    double s[3], r[3], x[3];
    double t1[3], t2[3], t3[3], l, theta, test;
    
    sscanf(argv[1], "%le\", &test);
    s[0]=1000.0; s[1]=0.0; s[2]=10.0;
    r[0]=0.0; r[1]=1.0; r[2]= 1.0;
    printf("x\ty\n");
    
    for( i=1; i<1000; i++ ) { for( j=100; j>=0; j-- ) {
        x[0]=(double)i; x[1]=(double)j/10.0; x[2]=0.0;
        l=0.0; for( k=0; k<3; k++ ) { t1[k]=x[k]-s[k]; l+=t1[k]*t1[k]; }
        l = sqrt(l); for( k=0; k<3; k++ ) t1[k]/=l;
        l=0.0; for( k=0; k<3; k++ ) { t2[k]=x[k]-s[k]; l+=t2[k]*t2[k]; }
        l = sqrt(l); for( k=0; k<3; k++ ) t2[k]/=l;
        l=0.0; for( k=0; k<3; k++ ) { t3[k]=0.5*(t1[k]+t2[k]); l+=t3[k]*t3[k]; }
        l = sqrt(l); for( k=0; k<3; k++ ) t3[k]/=l;
        theta = acos(t3[3])/DTOR;
        if( (theta<=test) || (x[0]==0.0) ) { printf("%f\t%f\n", x[0], x[1]); break; }
    }
    exit(0);
}

map view

For a light ray to reflect off a point \( x = (x, y, 0) \) on the lake, the surface normal at that point on the lake must have the right orientation for Snell's law to be satisfied. The surface normal is the average of the incident and reflected ray directions. If \( t_3 \) is inclined from vertical more than the rippling allows, then that point cannot reflect light. So find the locus of points on the lake surface where the normal is inclined less than some specified amount. This region is lit up.
With \( \frac{4}{3} \) thinned in degrees of elevation.

Focus of Light

Envelope of...