A simple algorithm for distances and worldlines in flat FLRW cosmology

M. G. Helland,[1★]

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
In situations where one would like to calculate distances or worldlines for objects in a flat, expanding universe there are many techniques available. The algorithm demonstrated here is discussed primarily for its sheer simplicity, relying on minimal knowledge of cosmological models. This would be of practical use where an accurate numerical representation of an expanding universe is required in contexts where the typical tools of a cosmologist would not be available.

Key words: galaxies: distances and redshifts

1 INTRODUCTION

Given a redshift z and a set of cosmological parameters (e.g. H_0, Ω_Λ, Ω_M) one can determine a comoving distance (d_C), an angular diameter distance (d_A), and a light travel time distance (d_T) associated with that redshift using the equations of FLRW cosmology. A common way of finding these distances, and the method implemented by various online cosmology calculators (e.g. Cappi 2001; Wright 2006), is to integrate over z for comoving distance:

$$d_C = \frac{c}{H_0} \int_0^z \frac{dz'}{E(z)} \tag{1}$$

Where:

$$E(z) = [\Omega_R (1+z)^4 + \Omega_M (1+z)^3 + \Omega_k (1+z)^2 + \Omega_\Lambda]^{1/2} \tag{2}$$

The angular diameter distance is then given by $d_A = d_C/(1+z)$.

The light travel time distance is given by a similar integral:

$$d_T = \frac{c}{H_0} \int_0^z \frac{dz'}{(1+z)E(z)} \tag{3}$$

An alternative method, presented here, involves computing d_A, d_T, and z simultaneously using an algorithm that works by tracking two photons traveling backwards in contracting space. This algorithm will be called the “distance algorithm” and is presented in Section 2. It is an abridged version of a second algorithm, presented in Section 3 and called the “worldline algorithm”, which tracks other objects in contracting space.

The algorithms are presented in Javascript due to the language’s ubiquity and accessibility. The algorithms apply to a flat FLRW ($\Omega_\Lambda + \Omega_M = 1$) cosmology.

2 THE DISTANCE ALGORITHM

```
var findZ = 10
var H0 = 70
var OmegaL = 0.7
var OmegaM = 1 - OmegaL

// convert km/s/Mpc  to  Mly/My/Mly
H0 = H0 / 3.08e19 * 60 * 60 * 24 * 365 * 1e6
var H = H0
var c = 1

// the photons, x1 has a head start
var x1 = 0.1
var x2 = 0

var t = 0, z = 0, data = []

while (z < findZ) {
    t--

    // move the photons
    x1 += c - H * x1
    x2 += c - H * x2

    // compare their separation
    z = 0.1 / (x1 - x2) - 1

    // update the Hubble parameter
    H = H0 * Math.sqrt(OmegaM *
        Math.pow(1 + z, 3) + OmegaL)

    data.push({
        z,
        d_A: x2,
        d_C: x2 * (1+z),
        d_T: -c * t
    })
}
```

Running the distance algorithm populates the data array, and the results can be seen in Fig. 1.

The algorithm has one loop, which decrements the time counter t and moves the photons. The unit of time is 1 million years and the unit of distance is 1 million light years. The distance coordinate is the proper distance from an observer at $x = 0$.

★ E-mail: mike@mikehelland.com

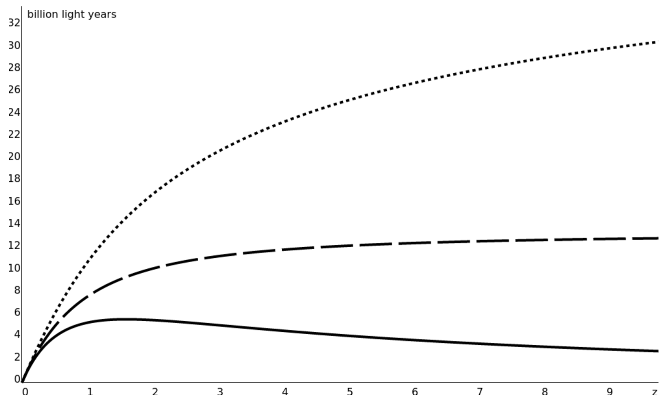

Figure 1. The results of the distance algorithm showing d_C (dotted), d_T (dashed), and d_A (solid) over z.

Since $t = 0$ is considered the present, and the photons x_1 and x_2 are at the origin, the algorithm begins with the photons being observed and has them traveling back to their source as it runs.

The photons travel at $c - Hx$, which mimics the effect of photons traveling through expanding space, but in reverse.

The photons at the observer are separated by 0.1 Mly. As they travel back in time we can compare the contracted separation to the original separation to determine a redshift, with:

$$1 + z = \frac{0.1}{x_1 - x_2} \tag{4}$$

After a new redshift has been calculated, the algorithm updates Hubble's parameter, H, for the next iteration. In this version of the algorithm $E(z)$ is simplified by omitting Ω_R and Ω_k.

Lastly, the data z, d_A, and d_T are logged, as well as $d_A(1 + z)$ as d_C .

Two photons are necessary to account for the evolution of Hubble's parameter. It may be of some interest that attempting to use a single photon, and calculating redshift using:

$$1 + z = \frac{c}{c - Hx} \tag{5}$$

produces accurate output for an exponentially expanding universe ($\Omega_\Lambda = 1$) with a static Hubble parameter, but not for models with matter ($\Omega_M > 0$)

The algorithm quits when it finds the target z, although it could just as easily be looking for a target t, or x, or $x(1 + z)$, or H. So the algorithm works just as well finding redshift for a distance as distances for a redshift.

3 THE WORLDLINE ALGORITHM

```
...
var t = 0, z = 0, data = [0]

// add galaxies to our model, 1 Bly apart
var objs = []
for (let i = 1000; i < 30000; i += 1000) {
    objs.push({var t = 0, z = 0, data = []
        i: objs.length,
        x: i,
        x0: i,
        data: [i] })
}

while (z < maxZ) {

    x1 += c - H * x1
    x2 += c - H * x2
    z = 0.1 / (x1 - x2) - 1
    t++

    data.push(x2)

    for (var i = 0; i < objs.length; i++) {
        if (objs[i].t) continue

        // move the galaxies
        objs[i].x -= H * objs[i].x
        objs[i].data.push(objs[i].x)

        // record when the photons reach them
        if (objs[i].x <= x2) {
            objs[i].t = t
            objs[i].z = z
        }
    }

    // update the Hubble parameter
    H = H0 * Math.sqrt(OmegaM *
        Math.pow(1 + z, 3) + OmegaL)
}
```

The setup code (the first 15 lines) for this algorithm are the same as the distance algorithm, indicated by "...".

This version of the algorithm places a galaxy every 1 billion light years from the observer at the beginning of the model, stored in `x0`. Since that represents their present proper distance, that is the galaxy's comoving distance. As the algorithm runs the galaxy's proper distance is updated according to Hubble's law, but in reverse. If the galaxy's velocity was Hx they would be moving away. Since they're going backwards through time, they should be getting closer together, so their velocity is $-Hx$.

At every time step, the distance of each galaxy is recorded, as well as the distance of the photon. This allows us to chart the worldlines of the galaxies, as shown in Fig 2.

When the photon meets a galaxy, the time and redshift are recorded and the galaxy stops moving, meaning its `x` represents its angular diameter distance, where the galaxy was when the light being observed today was emitted. Its comoving distance is given by the variable `x0`, which was set in the initial conditions. The distances in this algorithm are reported independently, unlike the distance algorithm which reports comoving distance (d_C) in terms of angular diameter distance ($d_A[1 + z]$).

The recessional velocity of each galaxy from the observer can be investigated by the results of the algorithm. Every time step has a distance for a galaxy, so its recessional velocity is $(x_t - x_{t-1})$ divided by the time step, which is 1 million years. The recessional velocity of a galaxy at the angular diameter distance turnaround (where the angular diameter distance stops increasing and starts decreasing) is shown in Fig 3. Its recessional velocity is c at t_{emit}, and then it slows down and speeds up again to just over c at $t_{observe}$, which is the present time. This tells the story of a universe that was decelerating and is now accelerating.

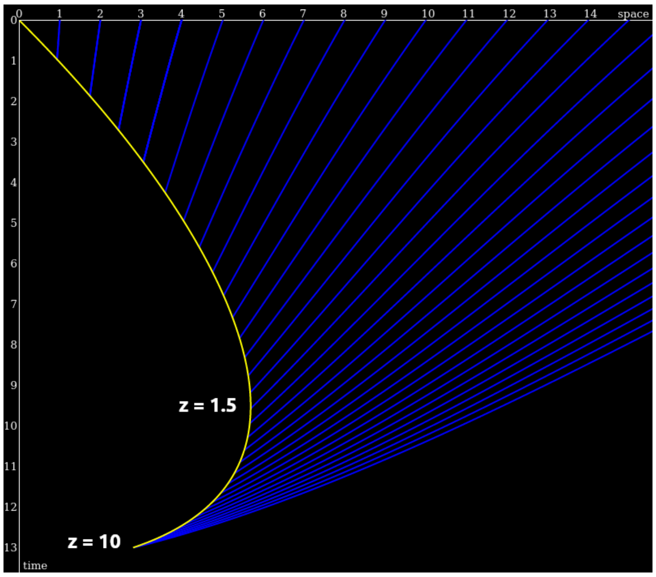

Figure 2. The results of the worldline algorithm showing the worldline of the photon in yellow, and the worldlines of galaxies that could have emitted that photon in blue.

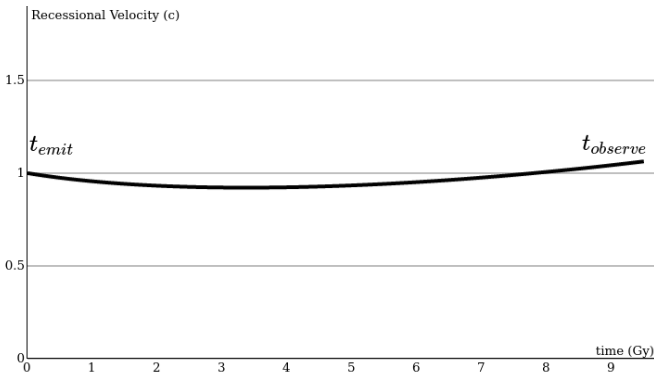

Figure 3. The recessional velocity of a galaxy that emitted light from the angular diameter distance turnaround, demonstrating the acceleration of the universe's expansion.

4 PRECISION

The precision of the algorithm is primarily limited by its time step, which is 1 million years in the versions presented. Using a smaller time step is as simple as converting `H0` to a smaller unit of time than 1 million years. For example, instead of `... 365 * 1e6` on line 7, use `... 365 * 1e4` for the time step to be ten thousand years. The distances reported will be in ten thousand light years too.

Using a time step fewer than a million years takes more time and uses more memory, of course. On an average, modern laptop this isn't an issue, but the effect on the results are negligible as well.

5 CONCLUSIONS

The algorithms presented here are intuitive in that $-Hx$ represents the Hubble flow in reverse that acts on all objects and photons. The use of two photons is necessary to compute the correct z value for a universe with a dynamic expansion rate.

The algorithm can be supplied as a target input one of any of the values it outputs, so that distances can be found by redshift, or redshift can be found if given a distance or time or expansion rate (H).

The simplicity of the algorithm may make it suitable in situations where the typical tools of a cosmologist aren't available. Perhaps to be implemented in presentation software, or educational materials.

The algorithm itself may be used to aid the teaching and investigation of an expanding universe to students with only basic math skills, without sacrificing accuracy.

The sparse nature of the algorithm, which computes its various outputs in-concert but can report them independently, may be of some value in facilitating the development of extensions, modifications, or alternatives to the current FLRW cosmology.

DATA AVAILABILITY

The data used in this work was generated by the included algorithms.

REFERENCES

Cappi A., 2001, Testing Cosmological Models with Negative Pressure (arXiv:astro-ph/0105382)
Wright E. L., 2006, PASP, 118, 1711