**Cosmological Kinematics**

**The Robertson-Walker Metric**

Now we’d like to solve for the universe’s metric, and calculate its evolution. To simplify matters, we’ll presume that we can adopt a time-like coordinate which is physical for every point in space, and which is moreover in the instantaneous rest frame of the matter present. I think this makes the spatial coordinates (r,θ,φ) stuck with the physical location. So like when you draw, say, a coordinate line on a balloon, the line expands as the balloon expands. In this way, the point on the balloon next to a given coordinate never moves w/r to it. It’s a common assumption that our universe is homogeneous and isotropic. Homogeneity implies things are the same at every point. So coordinates are indistinguishable. This still allows for homogeneous things like a uniform electromagnetic field blanketing the entire universe, pointing in the ‘z’ direction. The assumption of isotropy rids us of preferred directions so that even this would not be possible. Under the physical time assumption, considering the GR geometry file, it is possible to go to a coordinate frame within which the metric would look like this:



where hij depend only on position. The isotropy assumption means everything looks the same in all directions. Let 0 go in the time-like direction, r go in the radial direction, θ in the theta directions, and φ in the phi direction. Then we could write the metric as:



The homogeneity condition puts requirements on b(r). But note that b(r) can be a complicated function of r, as you can recall the metric for the sphere surface is homogeneous (see Tensor notes) even while being a function of r when expressed in polar coordinates. One homogeneity condition would be that the curvature be the same at every point (though possibly a function of time). This will turn out to be sufficient to determine the form of b(r). So let’s work out Rαβ in general, then the curvature, and see what b(r) must be. So we’ll write the metric tensor as:



where T = -1, R = a2(t)b(r), Θ = a2(t)r2, Φ =a2(t)r2sin2θ, and e00, e11, e22, e33 are just place holders indicating the position of the elements in the matrix. Proceeding,



and (*subscripts indicate partial derivatives w/r to that variable*)



and so we should have for Γ,



The Ricci tensor is:



Now taking the derivative and trace at the same time, to form the first term in the Ricci tensor we have:



Next,



where in the last line we’ve used knowledge of the product form of our terms. Next,



and last,



So all together,



Grouping together, noting identical terms,



this simplifies to:



Now let’s use fact that fct/f = 2aact/a2 = 2act/a, for any of the functions, f. Remember subscripts w/r to a variable indicate partial differentiation w/r to that variable, for brevity.



which simplifies to:



Let’s fill in T = -1, R = a2b, Θ = a2r2, Φ = a2r2sin2θ. Then we’ll have:



which simplifies to:



and,



and,



The curvature would be:



Now homogeneity would at least imply the curvature be independent of position, which means the last three terms must add to a constant. So we can say, choosing our constant wisely:



Let’s say u(r) = 1/b(r). Then,



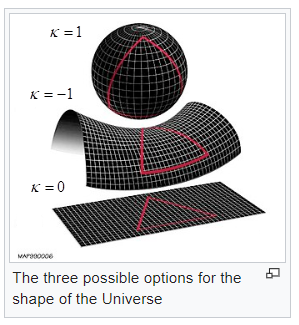
Now we must set λ = 0 since the metric must be regular everywhere. So we have:



So then our metric is so far,



This is called the Robertson-Walker metric. It’s noted that we may in complete generality consider just three values of κ: -1 (closed), 0 (flat), 1 (open), because we can change the scale of r to reduce any other value of κ to one of these values.



Note that *any* change of coordinates xα´ = fα´(xα) would be *equivalent* to this metric as well, just as any change of coordinates on a Euclidean metric is still Euclidean. So a physical-time, spatially homogeneous/isotropic universe doesn’t have to take this *explicit* form. Nonetheless, such a coordinate frame would be the preferred one to take. Now random question: how would you know if universe is closed, like a loop? Would metric tell you? I guess it would, just like the example in the tensor folder. We can use polar coordinates to describe a flat disk, as well as to describe the surface of a sphere. And it is the metric which tells us which of the two we’re dealing with. Anyway, the metric tensor is so far:



And the Ricci curvature tensor simplifies a bit. We’ll note that br/b2 = 2κr. So,



which we can write as:

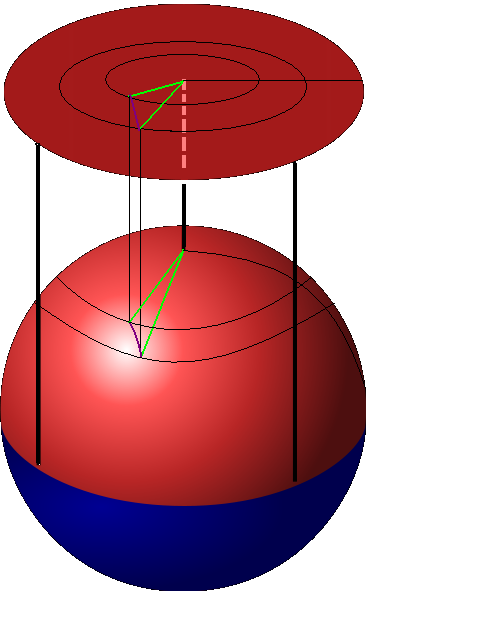


and the curvature, for what it’s worth, is (once again, subscript indicates partial derivative w/r to that variable, to condense notation a bit)



**The Hubble Expansion**

Let’s consider a formula for the distance between two objects at book-keeping coordinates (r1, θ1, φ1), (r2, θ2, φ2) respectively.



First, we can reposition the origin of our coordinate system to the first point, and reorient it so that the two points lie along the line θ = 0. Then the new points will be at radial coordinates r1 = 0 and r2 = ‘r’, respectively. This is completely general since we can place the origin and orientation of our coordinate system anywhere, and anyway, since the universe is homogeneous and isotropic. So then the physical distance would be:



which we can write as:



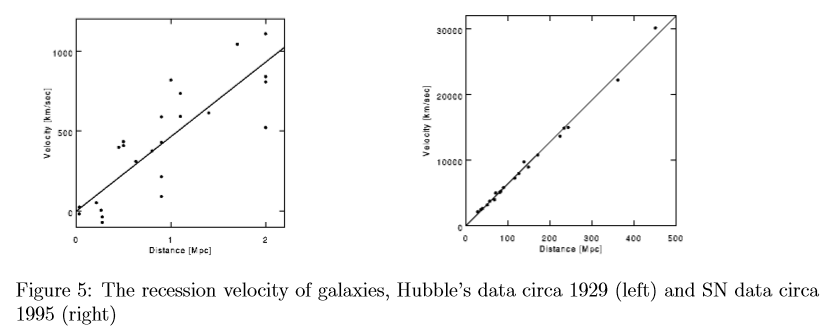
The recessional velocity between the two points would be:



Thus we come to Hubble’s law of sorts,



There is no indication here that H ought to be a constant. Nonetheless it appears to be:



Best present estimates are H = 67 km/s/Mpc, where one parsec is about 3.26 light-years.

**Redshift**

Let’s look at the redshift of light in an expanding universe. For this we need to know the geodesic of light. We have, quite generically:



And the Lagrangian is:



Let’s look at the equation of motion of *t*. So,



But we also have:



Plugging this into our dynamical equation for d2t/dλ2, we have:



Pretty ugly, but note we can write this as:



And so we have:



Now we can say that the wavelength is proportional to the time elapsed per unit affine parameter λ (unfortunately, same notation as wavelength λ). So wavelength follows proportionality:



Therefore we can compare wavelengths at at two different times:

