Antimatter Annihilation in the Early Universe

A. B. Boyd

Advisor: T. A. Moore

Pomona College

May 4, 2011

1 Introduction

One of the greatest mysteries of modern physics comes from the simple observation that matter exists in decent quantities in our surroundings. To be more explicit, it is strange that for all the matter in our surroundings, which is mostly composed of baryons, we don't see a corresponding amount of antimatter, which is composed of antibaryons. We can be very certain of the fact that there aren't many anti-baryons in our surroundings, because the reaction between a baryon and its corresponding anti-particle turns the sum of their masses into energy, which is detectable. This effect is so significant that we can be almost certain that there aren't significant amounts of antimatter in the observable universe, because we would be able to observe the highly energetic radiation that would come from the interaction with matter. Thus, it seems that our entire universe is composed of baryons. However, given our current understanding of the universe, the primacy of matter in the universe is an unexplained phenomenon.

The modern theory of the beginning of the universe states that the universe began with a Big Bang. Everything was very hot and very dense in the beginning, but the universe expanded and cooled. It has taken 13.7 billion years of cooling to arrive at our current state, where the average temperature of the universe is 2.7 Kelvins, but until 10^{-12} seconds after the Big Bang, the universe was still so hot that all reactions between the different types of fundamental particles in the universe were kinematically allowed.[1] Essentially, photons were so energetic that they could react to create massive particles, such as baryons and anti-baryons. Additionally, because the universe was so hot, the reactions were in equilibrium. The relative densities of particles maintained a constant proportionality with respect to each other during this epoch, because in the high temperature limit, their number densities were all proportional to T^3.[3] Thus, if the reactions were truly symmetric with respect to any particle and its antiparticle, the number densities of baryons and antibaryons would be equivalent while equilibrium was maintained. Given this initial condition of the universe, and the symmetries of all the interactions we've observed, the current asymmetry between baryons

and antibaryons requires explanation.

The process which generated the baryon-antibaryon asymmetry we see today is called baryogenesis. However, there is not a unique process which is attached to the name: there is actually a multitude of proposed mechanisms.[2] With so many potential candidates for the process, the actual process remains very mysterious. However, in 1967, A. Sakharov simplified the problem by proving that any theory of baryogenesis needs to include three key elements to be viable.[2] The first condition is that there must be a violation of baryon number conservation. Baryons have a baryon number equal to one, and antibaryons have a baryon number equal to minus one, so the the symmetry between baryons and antibaryons in particle interactions is equivalent to baryon number conservation. Thus, this symmetry must be violated. The second condition is that the process must violate charge and charge-parity symmetry. This basically means that the forces won't interact in the same way with a particle and its antiparticle, or with two oppositely charged particles. This can result in different reaction rates for oppositely charged particles, and for particle-antiparticle pairs. The third condition is that the process must include a departure from equilibrium.[4] These three conditions form the foundation for our project.

Our goal is to computationally analyze a toy model for baryogenesis, so that we might better understand the dynamics of the process. Specifically, we will be doing an n-body simulation where we track the position, energy, and momentum of each particle for a sequence of discrete timesteps. We will also develop a model consistent with the Sakharov constraints.

One of our main goals in developing this model is simplicity. Because of the very abstract nature of our model, any additional degrees of freedom we introduce will also make the parameter space that we need to consider much larger. In the interest of simplicity, we will limit our model to fewer particles than there are in reality. We will say the universe is composed entirely of a single type of baryon, which we will label b, and its corresponding antibaryon $\bar{b}$. For the purposes of the model, this baryon will be uncharged, but in spite of this only interacts through electromagnetic forces. Therefore, we also need to include photons in our model, which are represented by γ. Then, we

need to consider exactly how these particles interact with each other. The main interaction in our simulation will be

$$\gamma + \gamma \leftrightarrow b + \bar{b}.$$

This interaction conserves all the internal quantities of our particles. Thus, in order to satisfy the first condition that A. Sakharov laid out, we will also propose that the particles also have a small chance of undergoing the interactions

$$\gamma + \gamma \leftrightarrow \bar{b} + \bar{b},$$

and

$$\gamma + \gamma \leftrightarrow b + b.$$

The essence of the second requirement outlined by Sakharov is that, without a violation of charge and charge/parity symmetry, the decay rates of baryons and their corresponding antiparticles would be equivalent, meaning that no net change in baryon number occurs. If our model posits different reaction rates for the second two interactions shown above, it will therefore satisfy the second Sakharov condition. This will be the extent of the possible interactions between particles, and all that remains is to consider the environment within which the particles exists.

The universe has been expanding and cooling for the past 13.7 billion years. This is a result of the evolving metric of the universe. As stated before, very early in the universe, the temperature was so high that all the possible interactions between baryons, antibaryons, and photons were happening constantly, keeping the densities of the particles proportional to each other, and the universe at thermal equilibrium. However, as the universe cools, the photons, baryons, and antibaryons lose energy. This does not create a problem for the interactions which generate photons, because photons have no chemical potential and no rest energy. But, if the photons fall below a certain energy, they can no longer interact to create baryons and antibaryons, because if the energy of the photon is below the rest mass energy of baryons or antibaryons, the interaction would not conserve energy.

Once this temperature is reached, the system falls out of equilibrium, because the $\gamma + \gamma \rightarrow b + \bar{b}$ completely stops, but the balancing reaction $b + \bar{b} \rightarrow \gamma + \gamma$ is still possible. The universe won't be in equilibrium until the rate of baryon-antibaryon annihilation matches the rate of creation, which is negligibly small. By considering the dynamics of the universe between these two equilibrium states, we satisfy the last Sakharov condition, which mandates that the process depart from equilibrium. Thus, our model meets all of the Sakharov conditions.

The hope would be that the final equilibrium state that is reached after the transition would be dominated by either baryons or antibaryons, reflecting the current state of the observed universe, and therefore validating the ideas behind this model as a possible mechanism. However, even if that is not the result, there are other possible gains by simulating this model. Creating the simulation would provide a computational basis for examining more complicated interaction schemes numerically. With such an open question, some numerical analysis may give some interesting leads.

2 Theory

Constructing an algorithm for handling particle interactions in our model requires understanding the metric of the universe. A homogenous and isotropic universe should have a metric equation of the form

$$ds^2 = -dt^2 + a^2[dr^2 + q^2(d\theta^2 + sin^2\theta d\phi^2)], \tag{1}$$

where $q = r$ for a flat spatial geometry, and a is a unitless value which depends only on time and indicates the scale of the universe.[1] r is a spatial coordinate which is comoving with the fluid of fundamental particles that fills space: the r coordinate of a particular fluid element remains constant as the universe expands. Thus the fluid element's physical distance separating fluid elements having a fixed coordinate difference in r increases in proportion to a with time. If $q = r$, then the spatial part of the metric equation becomes recognizable as $dr^2 + r^2(d\theta^2 + sin^2\theta d\phi^2) = dx^2 + dy^2 + dz^2$,

so our metric equation can be written

$$ds^2 = g_{uv}dx^u dx^v = -dt^2 + a^2[dx^2 + dy^2 + dz^2]. \tag{2}$$

The metric tensor can be written as the matrix

$$g_{uv} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & a^2 & 0 & 0 \\ 0 & 0 & a^2 & \\ 0 & 0 & 0 & a^2, \end{bmatrix} \tag{3}$$

and the inverse metric can be written

$$g^{uv} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & a^{-2} & 0 & 0 \\ 0 & 0 & a^{-2} & \\ 0 & 0 & 0 & a^{-2}. \end{bmatrix} \tag{4}$$

It turns out that the time-evolution of a in this metric is determined by the differential equation

$$\left(\frac{1}{H_0}\frac{da}{dt}\right)^2 = \Omega_k + \frac{\Omega_m}{a} + \frac{\Omega_r}{a^2} + \Omega_v a^2, \tag{5}$$

where Ω_k is a measure of the curvature of space, which is zero for our universe, and Ω_m, Ω_r, and Ω_v are constants corresponding to the present ratios of mass energy density, radiation energy density, and vacuum energy density, respectively, to the critical energy density required for flat spatial geometry at the current time. H_0 is the currently observed value of the Hubble Constant. Objects in our universe move away from each other at velocities which are an increasing function of the distances between them. H_0 determines the functional relationship between distance and speed. For a particular galaxy, we can determine its distance d from us, and the speed v with which it is moving away from us, using the Doppler shift of its light, and this allows us to solve for the Hubble Constant, and gives us a fully determined differential equation. All that is left is to determine the initial condition.

We define the value of $a = 1$ at current time, and we know from the differential equation that a has increased radically over the lifetime of the universe. Thus, we know that a was very small at the time when the particles in the universe fell out of equilibrium with the photon gas. This period, when a was very small, is known as the radiation era, because small values of a cause the $\frac{\Omega_r}{a^2}$ term to dominate in our differential equation. This allows us to simplify the equation to

$$\left(\frac{1}{H_0}\frac{da}{dt}\right)^2 = \frac{\Omega_r}{a^2}. \tag{6}$$

This form of the equation is solvable, and gives us an analytic expression for a,

$$a(t) = \sqrt{a_i^2 + 2tH_0\sqrt{\Omega_r}}, \tag{7}$$

where a_i is our initial value of a, and t is the time that has passed since that initial state. This expression is valid as long as the the initial value of a_i, as well as $a(t)$, occurred in the radiation era.

The metric also allows us to determine the motion of the massive particles in our simulation using the geodesic equation

$$\frac{d\mu^\alpha}{d\tau} = -\Gamma^\alpha_{\beta\mu} u^\beta u^\mu, \tag{8}$$

because the Christoffel symbols $\Gamma^\alpha_{\beta\mu}$ are determined by the metric. We can use this to determine how the four-velocity u^α changes with coordinate time t. The Christoffel symbols are defined by

$$\Gamma^\alpha_{uv} = \frac{1}{2} g^{\alpha\nu} \left(\partial_u g_{v\nu} + \partial_v g_{\nu u} - \partial_\nu g_{uv}\right) \tag{9}$$

Here, $\partial_x y$ represents $\partial y/\partial x$. The only nonzero Christoffel symbols for our metric are

$$\Gamma^t_{xx} = (\partial_t a)a, \tag{10}$$

$$\Gamma^x_{xt} = \Gamma^x_{tx} = (\partial_t a)a^{-1} \tag{11}$$

$$\Gamma^y_{yt} = \Gamma^y_{ty} = (\partial_t a)a^{-1} \tag{12}$$

$$\Gamma^z_{zt} = \Gamma^z_{tz} = (\partial_t a)a^{-1}. \tag{13}$$

Using the fact we can parameterize the particle path with the coordinate time, we can write $\frac{du^\alpha}{d\tau} =$

$\frac{du^\alpha}{dt}\frac{dt}{d\tau} = \frac{du^\alpha}{dt}u^t$. This, along with equation (8) gives us the following equations

$$\frac{du^t}{dt}u^t = -a(\partial_t a)z(u^{x2} + u^{y2} + u^{z2}) \tag{14}$$

$$\frac{du^x}{dt}u^t = -2(\partial_t a)a^{-1}u^x u^t \tag{15}$$

$$\frac{du^y}{dt}u^t = -2(\partial_t a)a^{-1}u^y u^t \tag{16}$$

$$\frac{du^z}{dt}u^t = -2(\partial_t a)a^{-1}u^z u^t. \tag{17}$$

This system of equations is analytically solvable. Note thate $\frac{du^x}{dt}u^t = -2(\partial_t a)a^{-1}u^x u^t$ can be rewritten as $\frac{du^x}{dt}\frac{1}{u_x} = -2(\partial_t a)a^{-1}$. Then, we can integrate both sides with respect to time from some arbitrary $time = t_0$ to current $time = t$

$$\begin{aligned}
&\int_{t_0}^{t}\frac{1}{u^x}\frac{du^x}{dt}dt = -2\int_{t_0}^{t}\frac{1}{a}\frac{da}{dt}dt \\
&\int_{u_0^x}^{u^x(t)}\frac{1}{u^x}du^x = -2\int_{a_0}^{a(t)}\frac{1}{a}da \\
&ln[u^x(t)] - ln[u_0^x] = -2(ln[a(t)] - ln[a_0])
\end{aligned} \tag{18}$$

where $a_0 = a(t_0)$ and $u_0^x = u^x(t_0)$. Analogous relations apply for the y and z coordinates. This equation allows us to easily solve for the time dependence of the spatial components of the four-velocity

$$(u^x(t), u^y(t), u^z(t)) = (u_0^x, u_0^y, u_0^z)\frac{a_0^2}{a(t)^2}. \tag{19}$$

Given the relationship above, we can also solve for time dependence of the time component. If we plug in the result above for u^x, u^y, and u^z, into equation (14) we get

$$\frac{du^t}{dt}u^t = -a(\partial_t a)\frac{a_0^4}{a^4}\left(u_0^{x2} + u_0^{y2} + u_0^{z2}\right). \tag{20}$$

We can integrate both sides from t_0 to t as before to get

$$\int_{u_0^t}^{u^t(t)} u^t du^t = -a_0^4\left(u_0^{x2} + u_0^{y2} + u_0^{z2}\right)\int_{a_0}^{a(t)}\frac{da}{a^3}, \tag{21}$$

where I have defined a_0 and u_0^t as before. Again, this equation is easily solvable for $u^t(t)$ by integration, yielding

$$\frac{1}{2}(u^t(t))^2 - \frac{1}{2}(u_0^t)^2 = -a_0^4\left(u_0^{x2} + u_0^{y2} + u_0^{z2}\right)\left(-\frac{1}{2a(t)^2} + \frac{1}{2a_0^2}\right) \tag{22}$$

$$u^t(t) = \sqrt{{u_0^t}^2 + \frac{a_0^4({u_0^x}^2 + {u_0^y}^2 + {u_0^z}^2)}{a(t)^2} - a_0^2({u_0^x}^2 + {u_0^y}^2 + {u_0^z}^2)}. \tag{23}$$

Technically, there is also a negative solution for u^t, but we know that the coordinate time t for a particle always increases with the proper time τ of the particle. Thus $u^t = dt/d\tau \geq 0$. Given an initial value of the four-momentum, we can use these equations to evolve the four-velocities of massive particles in our system.

It should also be noted that the four momentum is defined for massive particles by

$$(E, p^x, p^y, p^z) = m\left(u^t, u^x, u^y, u^z\right), \tag{24}$$

where m is the particle's invariant mass. For that reason, the four-momentum satisfies very similar equations. The equation for time evolution of the spatial components is

$$(p^x, p^y, p^z) = (p_0^x, p_0^y, p_0^z)\frac{a_0^2}{a^2}, \tag{25}$$

and the time evolution of the time component is

$$E = \sqrt{E_0^2 + \frac{a_0^4({p_0^x}^2 + {p_0^y}^2 + {p_0^z}^2)}{a^2} - a_0^2({p_0^x}^2 + {p_0^y}^2 + {p_0^z}^2)}. \tag{26}$$

However, for massless particles, the derivation for the four-momentum using the Christoffel symbols doesn't work, because $d\tau = 0$. Additionally, equation (24) doesn't hold, because we know that massless particles have non-zero four-momentum. Instead, we can take the limit as mass goes to zero in our equations for massive particles to get the behavior of the four-momentum for massless particles. When considering the time dependence of the spatial elements of the four-momentum, the zero mass limit doesn't actually change anything, because mass does not appear in that equation. However, for the time component of the four momentum, E, there is a possible simplification. In order to simplify, note that for our particular metric, the following relationship between mass, energy, and momentum applies:

$$-g_{uv}p^u p^v = E^2 - a^2\left({p^x}^2 + {p^y}^2 + {p^z}^2\right) = E^2 - a^2|p|^2 = m^2, \tag{27}$$

where $|p|$ is the magnitude of the spatial components of the four-momentum. Thus, the equation for energy becomes

$$\begin{aligned} E &= \sqrt{E_0^2 + a_0^2|p_0|^2\left(\frac{a_0^2}{a^2} - 1\right)} \\ &= \sqrt{E_0^2 - a_0^2|p_0|^2 + \frac{a_0^4}{a^2}|p_0|^2} \\ &= \sqrt{m^2 + \frac{a_0^4}{a^2}|p_0|^2}. \end{aligned} \tag{28}$$

It is easy to take the zero-mass limit for this expression, because $E^2 = a^2|p|^2$ for photons, yielding

$$\begin{aligned} E &= \sqrt{0 + \frac{a_0^2}{a^2}E_0^2} \\ &= \frac{a_0}{a}E_0. \end{aligned} \tag{29}$$

Thus, we know how the four-momentum of both photons and massive particles in our simulation evolve with coordinate time. These equations also tell us how the particles' positions evolve with coordinate time, because

$$\left(\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}\right) = \left(\frac{u^x}{u^t}, \frac{u^y}{u^t}, \frac{u^z}{u^t}\right) = \left(\frac{p^x}{E}, \frac{p^y}{E}, \frac{p^z}{E}\right). \tag{30}$$

Thus, we know how non-interacting particles in our system behave.

Any relativistic interaction must obey one particular rule, which is that the total four momentum of the system is conserved. In our particular model, we are only considering interactions which have two reacting particles and two produced particles. Thus, given the reaction $A + B \rightarrow C + D$, conservation of four-momentum requires that

$$\begin{bmatrix} E_A + E_B \\ p_A^x + p_B^x \\ p_A^y + p_B^y \\ p_A^z + p_B^z \end{bmatrix} = \begin{bmatrix} E_C + E_D \\ p_C^x + p_D^x \\ p_C^y + p_D^y \\ p_C^z + p_D^z \end{bmatrix} = \begin{bmatrix} E \\ p^x \\ p^y \\ p^z \end{bmatrix} \equiv p^\alpha. \tag{31}$$

Firstly, given p^α, we can determine whether or not the reaction is kinematically possible. Unless

$$-g_{uv}p^u p^v \geq (m_C + m_D)^2 \tag{32}$$

there isn't enough energy for the interaction to happen. This is because the lowest possible energy for two particles of mass m_C and m_D is given in the case where neither of them are moving in the rest frame, meaning $|p_C| = |p_D| = 0$, so $E = E_C + E_D = \sqrt{a^2|p_C|^2 + m_C^2} + \sqrt{a^2|p_D|^2 + m_D^2} = m_C + m_D$. Once you've determined whether or not the reaction will happen, the next step is to determine the four-momenta of C and D. This is much easier to do in the frame where the center of mass is at rest. In this frame, the sum of the momenta is zero, so $|p_C| = |p_D|$, and the time component of the total four-momentum is $p^t = E_A + E_B = E_C + E_D$. Given that $-g_{uv}p^u p^v = E^2 - a^2|p|^2 = m^2$, we find that $E_A + E_B = E_C + E_D = \sqrt{|p_D|^2 a^2 + m_D^2} + \sqrt{|p_C|^2 a^2 + m_C^2}$. Because $|p_C| = |p_D|$, this becomes

$$E_A + E_B = \sqrt{|p_C|^2 a^2 + m_D^2} + \sqrt{|p_C|^2 a^2 + m_C^2}. \tag{33}$$

The momentum $|p_C| = |p_D|$ is explicitly solvable from this expression by squaring both sides, yielding

$$(E_A + E_B)^2 = |p_C|^2 a^2 + m_D^2 + |p_C|^2 a^2 + m_C^2 + 2\sqrt{\left(|p_C|^2 a^2 + m_D^2\right)\left(|p_C|^2 a^2 + m_C^2\right)}. \tag{34}$$

Then, you isolate the square root term to get

$$(E_A + E_B)^2 - 2|p_C|^2 a^2 - m_D^2 - m_C^2 = 2\sqrt{\left(|p_C|^2 a^2 + m_D^2\right)\left(|p_C|^2 a^2 + m_C^2\right)}. \tag{35}$$

Again, square both sides to get

$$\left((E_A + E_B)^2 - 2|p_C|^2 a^2 - m_D^2 - m_C^2\right)^2 = 4\left(|p_C|^2 a^2 + m_D^2\right)\left(|p_C|^2 a^2 + m_C^2\right). \tag{36}$$

This becomes

$$(E_A + E_B)^4 + m_C^4 + m_D^4 + 2m_C^2 m_D^2 - 2(E_A + E_B)^2(2|p_C|^2 a^2 + m_D^2 + m_C^2) = 4m_C^2 m_D^2, \tag{37}$$

which can be solved for $|p_C|$ to yield

$$|p_C| = \frac{\sqrt{(E_A + E_B)^4 + m_C^4 + m_D^4 - 2(E_A + E_B)^2 m_C^2 - 2(E_A + E_B)^2 m_D^2 - 2m_C^2 m_D^2}}{2a(E_A + E_B)}. \tag{38}$$

The equations are spherically symmetric in the center-of-mass frame, so we can arbitrarily choose a direction for $\vec{p_C}$ in the center-of-mass frame from a spherically uniform distribution. Then, from

conservation of momentum, $\vec{p_D} = -\vec{p_C}$ in the center-of-mass frame. Then, to get the time component of the four-momentum in the center-of-mass frame, you use $E^2 = a^2|p|^2 + m^2$. Finally, to get the four-momentum in our original coordinate system, you use the inverse of the original transform used to get to the center-of-mass frame.

In order to get the transform to and from the center-of-mass frame, we need to solve for ${T^u}_v$ in the equation

$$p^u = {T^u}_v {p_{CM}}^v. \tag{39}$$

However, ${T^u}_v$ is a 4×4 matrix with sixteen entries, and the equation represents four linear equations, so the space of possible solutions for ${T^u}_v$ is extremely large unless we have general form for the transform that has much fewer than sixteen degrees of freedom. In the special case of the flat space-time metric $g_{uv}dx^u dx^v = -dt^2 + dx^2 + dy^2 + dz^2$, the transform becomes the Lorentz Tranform, which has the general form

$${L^u}_v = \begin{bmatrix} \gamma & -v_x\gamma & -v_y\gamma & -v_z\gamma \\ -v_x\gamma & 1+(\gamma-1)\frac{v_x^2}{v^2} & (\gamma-1)\frac{v_xv_y}{v^2} & (\gamma-1)\frac{v_xv_z}{v^2} \\ -v_y\gamma & (\gamma-1)\frac{v_yv_x}{v^2} & 1+(\gamma-1)\frac{v_y^2}{v^2} & (\gamma-1)\frac{v_yv_z}{v^2} \\ -v_z\gamma & (\gamma-1)\frac{v_zv_x}{v^2} & (\gamma-1)\frac{v_zv_y}{v^2} & 1+(\gamma-1)\frac{v_z^2}{v^2} \end{bmatrix}, \tag{40}$$

where $\gamma = \frac{1}{\sqrt{1-v^2}}$, $v^2 = v_x^2 + v_y^2 + v_z^2$, and v_x, v_y, v_z are the three degrees of freedom, which represent the velocity components of the reference frame we are transforming to with respect to the old frame. However the the Lorentz Transform doesn't work with the metric we are using, because it doesn't satisfy the key aspect of these coordinate transforms, which is that they preserve the values of scalars, such as mass, which are frame invariant. Essentially, this means that for some four-vector $A^\alpha = (A^t, A^x, A^y, A^z)$, the two quantities $g_{uv}A^uA^v$ and $g_{uv}{L^u}_a A^a {L^v}_b A^b$ are not necessarily equal if ${L^u}_v$ is the Lorentz Transform.

Instead, we need to find a different general form for a transformation in our new metric from one frame to another. Appendix A lays out the steps for finding the general form for a transformation

T, which takes us to a rest frame. To determine our specific T, we want it to solve

$$\begin{bmatrix} E \\ p^x \\ p^y \\ p^z \end{bmatrix} = T^{-1} \begin{bmatrix} E' \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{41}$$

where $-E'^2 = -E^2 + a^2(p^{x2} + p^{y2} + p^{z2})$. Appendix A tells us that if we define $\gamma = \frac{1}{\sqrt{1-a^2v^2}}$, and $v = \sqrt{v^{x2} + v^{y2} + v^{z2}}$, where v^x, v^y, and v^z are the components of the coordinate velocity, then the transforms T^{-1} and T can be written

$$T^{-1} = \begin{bmatrix} \gamma & \gamma v a^2 & 0 & 0 \\ \gamma v^x & \frac{\gamma v^x}{v} & \frac{-\sqrt{v^{y2}+v^{z2}}}{v} & 0 \\ \gamma v^y & \frac{\gamma v^y}{v} & \frac{v^x v^y}{v\sqrt{v^{y2}+v^{z2}}} & \frac{-v^z}{\sqrt{v^{y2}+v^{z2}}} \\ \gamma v^z & \frac{\gamma v^z}{v} & \frac{v^x v^z}{v\sqrt{v^{y2}+v^{z2}}} & \frac{v^y}{\sqrt{v^{y2}+v^{z2}}} \end{bmatrix}, \tag{42}$$

and

$$T = \begin{bmatrix} \gamma & -\gamma v^x a^2 & -\gamma v^y a^2 & -\gamma v^z a^2 \\ -\gamma v & \frac{\gamma v^x}{v} & \frac{\gamma v^y}{v} & \frac{\gamma v^z}{v} \\ 0 & \frac{-\sqrt{v^{y2}+v^{z2}}}{v} & \frac{v^x v^y}{v\sqrt{v^{y2}+v^{z2}}} & \frac{v^x v^z}{v\sqrt{v^{y2}+v^{z2}}} \\ 0 & 0 & \frac{-v^z}{\sqrt{v^{y2}+v^{z2}}} & \frac{v^y}{\sqrt{v^{y2}+v^{z2}}} \end{bmatrix}. \tag{43}$$

Thus, our matrix equation yields the four following equalities:

$$E = \gamma E' \tag{44}$$

$$p^x = \gamma v^x E' \tag{45}$$

$$p^y = \gamma v^y E' \tag{46}$$

$$p^z = \gamma v^z E'. \tag{47}$$

We can solve this system for v^x, v^y, and v^z, yielding

$$v^x = \frac{p_x}{E} \tag{48}$$

$$v^y = \frac{p_y}{E} \tag{49}$$

$$v^z = \frac{p_z}{E}. \tag{50}$$

In turn, this explicitly specifies the transform to the center-of-mass frame and back again.

If we know the reaction $A + B \rightarrow C + D$ happens, and we know the four-momenta of A and B and the masses of C and D, the algorithm for the collision can be described simply. Sum the four-momenta of the two reactants, and transform the sum to the center-of-mass frame. Once in the new frame, solve for the spatial momentum magnitude for product C. Then, choose a random direction from a spherically symmetric direction distribution for the direction of C's momentum. Given $\vec{p_C}$, conservation of momentum requires that $\vec{p_C} = -\vec{p_D}$. Finally, transform the four-momenta back to their original frame, using the inverse transform T^{-1}.

For the purposes of this model, in order for two particles interact with each other, three things must be true. One is that the interaction be one of the possible interactions given by $\gamma + \gamma \leftrightarrow \bar{b} + b$, $\gamma + \gamma \leftrightarrow \bar{b} + \bar{b}$, or $\gamma + \gamma \leftrightarrow b + b$, which were described earlier. The second condition is that the reaction is kinematically allowed, as described earlier, which means that the rest mass of the products must be accounted for in the four-momentum of the reactants. The last condition is that they are within a certain distance of each other, which I will call the interaction radius R. For the sake of simplicity, we will consider R to be independent of energy. This interaction radius is defined differently for each of the three fundamental reactions, such that reactions in either of the possible directions for each fundamental reaction have the same interaction radius. Since the radius of interaction determines the cross-section, which determines the reaction rate, having different interaction radii for $\gamma + \gamma \leftrightarrow \bar{b} + \bar{b}$ and $\gamma + \gamma \leftrightarrow b + b$ is necessary to satisfy the second Sakharov condition.

It should be noted that if we consider the positions of our of the particles in our current coordinate

system, because the distance between two points in the system is defined by

$$ds^2 = -dt^2 + a^2\left(dx^2 + dy^2 + dz^2\right), \tag{51}$$

where dx^α is the difference in the coordinate positions of the particles, the condition for reaction becomes a little bit more complicated. If the physical interaction radius for a reaction is R, then the particles must satisfy

$$ds \leq R. \tag{52}$$

Since we are only considering particles that interact at the same coordinate time, the relation between the difference in coordinate position and the distance between the particles is

$$ds = a\sqrt{dx^2 + dy^2 + dz^2}. \tag{53}$$

Thus, the condition for two particles with interaction radius R interacting is

$$\sqrt{dx^2 + dy^2 + dz^2} = \frac{ds}{a} \leq \frac{R}{a}. \tag{54}$$

3 Constructing the Simulation

For the purposes of testing the algorithm, I am programming in Mathematica 6. We will begin with a state of our system. That state will be represented by a list. At the very highest level, there will be three sublists of that list. The first sublists is a list that specifies the states of every baryon b in the simulation. The second sublist is a list that specifies the states of every antibaryon $\bar{b}$ in the simulation. The third sublist is a list that specifies the states of every photon γ in the system. Each of these three sublists has one element for every one of their respective particles in the system. For instance, the third list has an element for every photon in the simulated universe. For every particle, the corresponding element specifies all of the pertinent information about the particle. The element list for a particle takes the form $\{E, p^x, p^y, p^z, x, y, z\}$, where E is the energy, the second through fourth items are the momentum components, and the fifth through seventh items are the coordinates of the particle's position.

The essence of the simulation is evolving this list that describes the state of the system. For the purposes of this model, this simply includes using the time dependence of $a(t)$ to evolve our four-momentum, using the four-momentum to evolve the position, and carrying out the reactions between particles. In fact, given an initial state of the system, and an initial value of the spatial value a_0, the process can broken into broken into a few steps.

Choose the number of iterations you want to go through, N, and this will determine the number of time steps taken in the calculation. However, we do not choose each time step such that they are constant. This is because, earlier in the universe, particles were more tightly packed, meaning that collisions and interactions happened more often, and equilibrium was achieved more rapidly. This means that, in order for our time step to allow all of the interactions that happen early in the universe, it needs to be much smaller than it needs to be later in the universe, when particles are more spread out, and interacting less frequently. Specifically, we choose the time step $\triangle t$ to scale as $a(t)$. Thus, if we define our initial change in time to be $\triangle t_0$, then our change in time after the mth time step is

$$\triangle t_m = \triangle t_0 \frac{a(t_m)}{a(t_0)}, \tag{55}$$

where t_0 is the initial time, and t_m is the time at the mth time step. This gives us the recursive relation for the time of the simulation,

$$t_{m+1} = t_m + \triangle t_0 \frac{a(t_m)}{a(t_0)}. \tag{56}$$

We choose this time step, because of the fact that the the coordinate velocity of the particles $v^x = p^x/E$ scales as approximately $1/a$. Thus, with our time step defined as above, the change in the coordinates of our particles for every time step should remain relatively constant as a increases.

For every time step, the iteration first involves evolving the four momentum. Let $(E_n, p_n^x, p_n^y, p_n^z)$ be the four-momentum after the nth time step. Using the relations for the energy and momentum given in equations (25) and (28), we get the recursive relations

$$(p_{n+1}^x, p_{n+1}^y, p_{n+1}^z) = (p_n^x, p_n^y, p_n^z) \frac{a(t_n)^2}{a(t_{n+1})^2}, \tag{57}$$

and

$$E_{n+1} = \sqrt{E_n^2 + a^2|p_n|^2 \left(\frac{a(t_n)^2}{a(t_{n+1})^2} - 1\right)}. \tag{58}$$

Then, we need to evolve the position. While the algorithm for determining the value of the four-momentum is not an approximation, because it is derived from analytic expressions, the expression for the change in position, given by equation (30), isn't solved analytically. Instead, we evolve the coordinate position numerically. To make the results accurate to second order, we start by defining

$$\begin{aligned} p^\alpha_{n+1/2} &= \frac{1}{2}(p^\alpha_{n+1} + p^\alpha_n) \\ E_{n+1/2} &= \frac{1}{2}(E_{n+1} + E_n). \end{aligned} \tag{59}$$

Then, we define the recursive relation for the position of a particle as

$$x^\alpha_{n+1} = x^\alpha_n + \triangle t_0 \frac{a(t_n)}{a(t_0)} \frac{p^\alpha_{n+1/2}}{E_{n+1/2}}. \tag{60}$$

The last step for every iteration is carrying out the collisions in the system by deleting the reacting particles from the state list, and adding the products of the reaction to the state list. The resulting state will be the state of the system at time $t_{n+1} = t_n + \triangle t_n$.

It should be noted that for the sake of simulating a finite system, we have certain boundary conditions for our system. Topologically, we will take our system to be toroidal, in that it can be thought of as a cube, with each edge connected to its opposite. To include this fact in the calculation, all I need to do is take the remainder of the coordinates divided by the width of the box. Given this, the function for iterating the position and momentum of a particle, given that it doesn't interact, is simple.

```
step[vec_, dt_, b_, a1_, a2_] :=
 Table[If[i == 1,
   Sqrt[vec[[1]]^2 +
     a1^2*(vec[[2]]^2 + vec[[3]]^2 + vec[[4]]^2) (a1^2/a2^2 - 1)],
   If[i <= 4, vec[[i]]*a1^2/a2^2,
```

```
Mod[vec[[i]] +
  dt*(vec[[i - 3]] + vec[[i - 3]]*a1^2/a2^2)/(vec[[1]] +
      Sqrt[vec[[1]]^2 +
        a1^2*(vec[[2]]^2 + vec[[3]]^2 + vec[[4]]^2) (a1^2/a2^2 -
           1)]), b]]], {i, Length[vec]}]
```

It simply takes the input "vec", which is the state vector of the particle $\{E, p^x, p^y, p^z, x, y, z\}$, the change in time that occurs "dt", the width of our box "b", the values of the spatial constant "a1" and "a2" at time t and t+"dt", respectively.

The generation of the collisions is the most difficult part of the simulation. There is a radius of interaction which corresponds to each of the possible reactions. For $\bar{b} + b \leftrightarrow \gamma + \gamma$ the radius is labeled R_{MA}, for $b + b \leftrightarrow \gamma + \gamma$, the radius is labeled R_{MM}, and for $\bar{b} + \bar{b} \leftrightarrow \gamma + \gamma$, the radius is labeled R_{AA}. For the purposes of this model, if two particles are within their respective interaction radius, the reaction happens. If not, the particles don't react. However, given some interaction radii, it is always possible that more than two particles interact. Thus, we need some sort of process to decide which of the possible reactions actually happen, because a particle can only undergo one interaction at a time, because we aren't considering higher order interactions in our model. For the purposes of our model, we will choose our interactions randomly.

To select our interactions randomly without introducing some unintended ordering effects is important. For this program, I create a list of all the possible interactions, with distinct element for every set of reactants and products. Then, I randomly select one of the interactions. Let's say this interaction is between particle A and particle B, and it goes to particles of type C and D. Then, I will save this interaction, and delete all other interactions that involve either particle A or particle B. Then, I will repeat the process, until no interactions are left. The result is a list of all the interactions that will occur in our current time step. Every element specifies the particles that will interact, their states, and the types of particle they will turn into. Given this list, we can subtract the reacting particles from our current system, and carry out the each of the mandated reactions,

and add the resulting particles to our system.

For each of the reactions $A+B \rightarrow C+D$, we carry out an algorithm for generating the states of C and D. The algorithm for generating the four-momentum is mostly contained in the theory section. We sum the four-momenta of A and B, and then transform to a frame where the center of mass is at rest. We use the matrix forms described in the theory to do so. Then, we explicitly solve for the absolute value of the the momentum of both C and D in this frame p. This determines the energy of C and D. Then, we an arbitrary direction for the momentum of C by selecting from a spherically symmetric distribution of unit vectors. This unit vector is constructed by choosing ϕ randomly from the interval $(0, 2\pi)$, and choosing $\theta = cos^{-1}(1-2x)$ where x is chosen randomly from the interval $(0, 1)$. The unit vector is $(cos(\phi)sin(\theta), sin(\phi)sin(\theta), cos(\theta))$, so $p_C = (pcos(\phi)sin(\theta), psin(\phi)sin(\theta), pcos(\theta))$, and by conservation of momentum $p_D = (-pcos(\phi)sin(\theta), -psin(\phi)sin(\theta), -pcos(\theta))$. Thus, we have our four momenta in the rest frame, and by transforming back we get the four-momenta for our particles in our original frame. The function shown below, takes in the sum of the four-momenta of A and B, the mass of C and D, and the current value of a, and outputs four momenta of C and B. The function "invTrans", which is used in the function determines the transformation from the comoving frame to the rest frame of the interaction.

```
Collision[E_, px_, py_, pz_, mc_, md_, a_] :=
 Module[{invTr = invTrans[E, px, py, pz, a],
   restAB = {{Sqrt[E^2 - (a^2)*(px^2 + py^2 + pz^2)]}, {0}, {0}, {0}},
    phi = 2*Pi*Random[], theta = ArcCos[1 - 2*Random[]]},
  ptot = Sqrt[
     restAB[[1]][[1]]^4 + mc^4 + md^4 -
      2 (mc^2*md^2 + restAB[[1]][[1]]^2*mc^2 +
         restAB[[1]][[1]]^2*md^2)]/(2*a*restAB[[1]][[1]]);
  restC = {{Sqrt[ptot^2*a^2 + mc^2]}, {ptot*Cos[phi]*
     Sin[theta]}, {ptot*Sin[phi]*Sin[theta]}, {ptot*Cos[theta]}};
```

```
restD = {{Sqrt[ptot^2*a^2 + mc^2]}, {-ptot*Cos[phi]*
    Sin[theta]}, {-ptot*Sin[phi]*Sin[theta]}, {-ptot*Cos[theta]}};
ourC = Dot[invTr, restC]; ourD = Dot[invTr, restD];
Return[{{ourC[[1]][[1]], ourC[[2]][[1]], ourC[[3]][[1]],
   ourC[[4]][[1]]}, {ourD[[1]][[1]], ourD[[2]][[1]], ourD[[3]][[1]],
    ourD[[4]][[1]]}}]
```

The rest of the current program, including comments, which are surrounded by (* *), is included in Appendix B. The majority of this work is dedicated to programming in the step which decides all of the interactions, and creates the new state, plus the new particles, minus the old ones.

4 Future Work

There are still a few key ideas which need to be implemented in the program. One key element that needs to be included is generating the position of the products in the reaction $A + B \rightarrow C + D$. The position of both C and D will be generated by considering the point the lies halfway between A and B, which we will call $x^{\alpha}_{AB} = (x_{AB}, y_{AB}, z_{AB})$. Then, we will choose the position of C and D to be $x^{\alpha}_{AB} + x^{\alpha}$ and $x^{\alpha}_{AB} - x^{\alpha}$, respectively, where the length of x^{α} is R_{CD}, which represents the interaction radius of the reaction between C and D. The direction of x^{α} is arbitrarily selected from a spherically symmetric distribution of unit vectors in the same way as for generating the new momentum vectors. This may seem unusual and unphysical, but it is important to prevent an effect that could arise given large interaction radii. If, given some time step, particles don't travel farther than the interaction radius, and if C an D start with coordinates that are close together, then C and D would be within each others' interaction radius when we evaluate the set of collisions for the next time step. This would mean that under certain conditions, whenever two particles A and B would interact in one time step, their products C and D would react in the next time step. This is a far more unphysical situation than creating an artificial separation between C and D. Additionally,

because we are looking for large-scale effects in our system, as long as the position variations caused by this algorithm are not large in comparison to the scale of the size of the box in which the particles are contained, the global behavior of our simulation shouldn't be radically different, except for more accurate reaction rates.

The most immediate piece of the calculation that needs to be completed is generating the initial state of the system. The first guideline is that the initial temperature T_0 must be greater than the critical temperature T_C at which the system transitions out of the state where photons, baryons, and antibaryons are in equilibrium. This means that the average energy of the system must be greater than the mass energy of baryons and antibaryons. Because the average kinetic energy of a particle in three dimensions is $\frac{3}{2}kT$, the condition for our initial temperature becomes

$$T_0 \gg T_C = \frac{2}{3k} Max[2m_b, 2m_a], \tag{61}$$

where m_b is the baryon mass, and m_a is the antibaryon mass. Given some chosen value of T_0, we can determine the initial value of a, because in our metric,

$$Ta = constant. \tag{62}$$

Thus, given that we know the current value of a is $a_f = 1$ and the current temperature is $T_f = 2.7K$, we can find the initial value of a from T_0 by using the equation

$$a_0 = \frac{a_f T_f}{T_0} = \frac{2.7K}{T_0}. \tag{63}$$

Thus, as shown before, knowing the initial value of a allows us to functionally determine the value of a for all time values in the simulation.

Once we know the initial temperature and initial value of a, we can generate our initial state. The principle for how we will generate this initial state is dependent on the known dynamics of the early universe. Instead of generating an initial state where the number of of photons, baryons, and antibaryons are in equilibrium, we generate a state of only photons. Since the reactions happens so quickly in the early universe, as we iterate the program, the photons should quickly equilibrate with

baryons and antibaryons. We simply need to make sure that the system equilibrates well before the temperature drops below the critical temperature, which was described in the previous paragraph.

Thus, given an initial temperature we can thermodynamically derive the number density of photons in that system, as well as the energy distribution of the photons. Using the entropy of a photon gas, one can derive that the number density of the photons is

$$\mu = \frac{N}{V} = 2.404 \times 8\pi \left(\frac{kT}{hc}\right)^3 = 60.422 \left(\frac{kT}{hc}\right)^3. \tag{64}$$

[3] Since the number density is proportional T^{-3} for all our particles, and the volume of our box is proportional to a^3, which is proportional to T^{-3}, the number of particles in our box is constant. We can explicitly determine that constant number N from the number density μ and the width b of our box, evaluated at the same time coordinate

$$N = \mu b^3. \tag{65}$$

For each of the N photons in our initial state, we go through a simple algorithm to determine their position and four-momentum. First, we randomly select the coordinates of each photon by randomly selecting three numbers, each lying in the interval $(0, b)$. Since our coordinates are bounded by $(0, b)$ in all directions, this randomly spreads the photons out through our box. Then, to determine the four-momentum for each photon, we use the the Planck spectrum, which tells us that the energy density per unit photon energy is

$$\rho(\epsilon) = \frac{8\pi\epsilon^3}{(hc)^3(e^{\epsilon/kT} - 1)}, \tag{66}$$

where ϵ is the photon energy.[5] We want to know what the number density per unit photon energy is, which can be obtained from dividing the Planck spectrum by the photon energy ϵ, yielding

$$\mu(\epsilon) = \frac{8\pi\epsilon^2}{(hc)^3(e^{\epsilon/kT} - 1)}. \tag{67}$$

We will use this distribution to randomly generate energies E for each photon, yielding photon energies with an appropriate spread. While I haven't yet explicitly determined how to do this for

this particular number density, the general algorithm for doing this is to normalize $\mu(\epsilon)$, to make it a probability distribution, and integrate the probability distribution to yield

$$p(\epsilon) = \frac{\int_0^\epsilon \mu(x)dx}{\int_0^\infty \mu(x)dx}. \tag{68}$$

Then, select random numbers y from the interval $(0, 1)$ and solve $p(\epsilon) = y$ for ϵ. You will get random values of photon energy ϵ which obey the appropriate distribution. Unfortunately, $p(\epsilon)$ is not derivable analytically, so we will have to solve for ϵ numerically. Because these are photons, the absolute value of the spatial momentum is explicitly determined by the energy as $|p| = E/a$. Thus, we determine the direction of the spatial components of the four-momentum by selecting a direction randomly from a spherically symmetric distribution. This is done in the same way as when generating the the four-momentum of the reactants in a collision.

5 Appendix A

In order to derive the appropriate general transform, which can take us to the coordinate frame in which our system's center of mass is at rest, we need it to make sure it solves

$$g_{uv}A^u A^v = g_{uv}T^u{}_a A^a T^v{}_b A^b, \tag{69}$$

where A^μ is an arbitrary four-vector. However, we don't need to start from nothing. Even though our metric is different from the flat space-time metric, it still has the same symmetry in the spatial coordinates. Any transformation from one frame to another frame which is moving with respect to the original can be simplified to a single dimension. The velocity vector may have x, y, and z components, but we can rotate our reference frame so that the velocity now only points along the new x' direction. Thus, the transformation will not affect the y' and z' components of the vector, and our problem is simplified from trying to find a transformation which is represented by a 4×4 matrix, and could have as many as 16 variables, to trying to find the 2×2 matrix which relates the x' and t' components of the rest frame to the x and t components of the frame in which the center

of mass is moving.

We can simply express the transformation as

$$\begin{bmatrix} dt' \\ dx' \end{bmatrix} = L \begin{bmatrix} dt \\ dx \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} dt \\ dx \end{bmatrix}. \tag{70}$$

The key equation that this must obey, as state before, is that $-dt^2 + a^2(dx^2 + dy^2 + dz^2) = g_{uv}dx^u dx^v = g_{uv}L^u{}_a dx^a L^v{}_b dx^b = -dt'^2 + a^2(dx'^2 + dy'^2 + dz'^2)$. Since $dy = dy'$, $dz = dz'$, $dt' = Adt + Bdx$, and $dx' = Cdt + Ddx$, this simplifies to

$$-dt^2+a^2dx^2 = -dt'^2+a^2dx'^2 = -A^2dt^2-B^2dx^2-2ABdxdt+a^2(C^2dt^2+D^2dt^2+2CDdxdt). \tag{71}$$

This gives us three equations relating A, B, C, and D

$$-1 = a^2C^2 - A^2 \tag{72}$$

$$a^2 = a^2D^2 - B^2 \tag{73}$$

$$0 = -AB + a^2CD. \tag{74}$$

If we treat C as our free variable, then the above equations solve to the following in terms of C

$$A = \sqrt{1 + a^2C^2} \tag{75}$$

$$B = a^2C \tag{76}$$

$$C = \sqrt{1 + a^2C^2}. \tag{77}$$

Then, in order to understand how C relates to the velocity difference between the frames we're transforming between, consider the scenario where we have an object that is moving in the x direction with coordinate speed v, and is at rest in the frame we are transforming to. This means that, if we consider the change in time and position coordinates of the object dt and dx in our original frame, and compare with change in the time and position coordinates dt' and dx' in our new frame, they will satisfy

$$dt' = Adt + Bdx \tag{78}$$

$$dx' = Cdt + Ddx. \tag{79}$$

However, because the particle is stationary in the new frame, $dx' = 0$, and because it is moving with coordinate speed v in our original frame, $dx = vdt$. Thus, the second equation becomes

$$0 = (Cv + D)dx. \tag{80}$$

Plugging in D's dependence on C into this equation, we get $D = A = \frac{1}{\sqrt{1-a^2v^2}}$, $C = \frac{-v}{\sqrt{1-a^2v^2}}$, and $B = \frac{-va^2}{\sqrt{1-a^2v^2}}$. Thus, the transform for getting to the frame that is moving in the x-direction is

$$L^u{}_v = \begin{bmatrix} \frac{1}{\sqrt{1-a^2v^2}} & \frac{-va^2}{\sqrt{1-a^2v^2}} & 0 & 0 \\ \frac{-v}{\sqrt{1-a^2v^2}} & \frac{1}{\sqrt{1-a^2v^2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{81}$$

To get the inverse transform, you simply replace v with $-v$.

To find the full transformation, all we need to do is find out the rotation transform we need to apply to our original frame in order to have the object, which is stationary in the rest frame, move along the x-direction in our rotated frame.

If you consider a two dimensional euclidean space, a rotation of your coordinate system in that space can be represented by the transformation matrix

$$R = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}, \tag{82}$$

where a and b are real numbers in the range $[-1, 1]$, and $a^2 + b^2 = 1$. If we are given some arbitrary vector,(v_1, v_2), and we want to find the rotation which maps that vector to a vector with the same length, but which only has an x component, then you need to solve the equation

$$\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{83}$$

for a and b, where $v = \sqrt{v_1^2 + v_2^2}$. The resulting transformation matrix is

$$\begin{bmatrix} \frac{v_1}{v} & \frac{v_2}{v} \\ -\frac{v_2}{v} & \frac{v_1}{v} \end{bmatrix}. \tag{84}$$

If we are considering such a rotation in space-time that has three spatial dimensions though, the procedure is a little bit different. Any rotation in three dimensions can be described as the product of two rotations around two principle axes of the coordinate system, which are described by the rotation matrix above. In our particular case, because we want to rotate a vector such that it lies completely in the x-direction, we can first rotate our vector around the x-axis, until it lies entirely in the xy plane. Then you can rotate it about the z-axis such that it lies entirely along the x-axis. The first rotation is

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{v_y}{v} & \frac{v_z}{v} \\ 0 & 0 & -\frac{v_z}{v} & \frac{v_y}{v} \end{bmatrix}, \tag{85}$$

where $v = \sqrt{v_y^2 + v_z^2}$, and it is applied to the vector (v_t, v_x, v_y, v_z), yielding the vector $(v_t, v_x, v, 0)$. The second rotation is

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{v_x}{v'} & \frac{v}{v'} & 0 \\ 0 & -\frac{v}{v'} & \frac{v_x}{v'} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{86}$$

where $v' = \sqrt{v_x^2 + v_y^2 + v_z^2}$, and it maps $(v_t, v_x, v, 0)$ to $(v_t, v', 0, 0)$. Thus, the full rotation to get to the appropriate coordinate system is given by

$$R = R_2 R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{v^x}{v} & \frac{v^y}{v} & \frac{v^z}{v} \\ 0 & \frac{-\sqrt{v^{y\,2} + v^{z\,2}}}{v} & \frac{v^x v^y}{\sqrt{v^{y\,2} + v^{z\,2}}\, v} & \frac{v^z v^x}{\sqrt{v^{y\,2} + v^{z\,2}}\, v} \\ 0 & 0 & \frac{-v^z}{\sqrt{v^{y\,2} + v^{z\,2}}} & \frac{v^{y\,2}}{\sqrt{v^{y\,2} + v^{z\,2}}} \end{bmatrix}. \tag{87}$$

If we apply this rotation to the system's total four-momentum, it will convert it to a system with total four-momentum whose spatial part lies entirely in the x-direction. To get the inverse rotation, we can simply take the transpose of the matrix: $R^T = R^{-1}$.

Thus, the full transformation to the center-of-mass frame is $T = L^u{}_\alpha R^\alpha{}_v = LR$, and the transformation back is $T^{-1} = R^{-1\,u}{}_\alpha L^{-1\,\alpha}{}_v = R^{-1}L^{-1}$. If we define $\gamma = \frac{1}{\sqrt{1-a^2v^2}}$, then the transform T^{-1} can be written

$$T^{-1} = \begin{bmatrix} \gamma & \gamma v a^2 & 0 & 0 \\ \gamma v^x & \frac{\gamma v^x}{v} & \frac{-\sqrt{v^{y\,2}+v^{z\,2}}}{v} & 0 \\ \gamma v^y & \frac{\gamma v^y}{v} & \frac{v^x v^y}{v\sqrt{v^{y\,2}+v^{z\,2}}} & \frac{-v^z}{\sqrt{v^{y\,2}+v^{z\,2}}} \\ \gamma v^z & \frac{\gamma v^z}{v} & \frac{v^x v^z}{v\sqrt{v^{y\,2}+v^{z\,2}}} & \frac{v^y}{\sqrt{v^{y\,2}+v^{z\,2}}} \end{bmatrix}, \tag{88}$$

and T can be written

$$T = \begin{bmatrix} \gamma & -\gamma v^x a^2 & -\gamma v^y a^2 & -\gamma v^z a^2 \\ -\gamma v & \frac{\gamma v^x}{v} & \frac{\gamma v^y}{v} & \frac{\gamma v^z}{v} \\ 0 & \frac{-\sqrt{v^{y\,2}+v^{z\,2}}}{v} & \frac{v^x v^y}{v\sqrt{v^{y\,2}+v^{z\,2}}} & \frac{v^x v^z}{v\sqrt{v^{y\,2}+v^{z\,2}}} \\ 0 & 0 & \frac{-v^z}{\sqrt{v^{y\,2}+v^{z\,2}}} & \frac{v^y}{\sqrt{v^{y\,2}+v^{z\,2}}} \end{bmatrix}. \tag{89}$$

6 Appendix B

The following text is the commented Mathematica program used for the calculation. It is currently unfinished, and it does not include the definition of the function "invTrans[E, px, py, pz, a]", but it is defined as the matrix T^{-1}, which was given in the theory section.

```
a[t_, a0_, H0_, \[CapitalOmega]r_] :=
 Sqrt[a0^2 + 2*t*H0*Sqrt[\[CapitalOmega]r]]


    (*This describes the evolution of \
the universe constant given that all of the terms are constant*)


Collision[E_, px_, py_, pz_, mc_, md_, a_] :=
 Module[{invTr = invTrans[E, px, py, pz, a],
```

```
   restAB = {{Sqrt[E^2 - (a^2)*(px^2 + py^2 + pz^2)]}, {0}, {0}, {0}},
    phi = 2*Pi*Random[], theta = ArcCos[1 - 2*Random[]]},
  ptot = Sqrt[
     restAB[[1]][[1]]^4 + mc^4 + md^4 -
      2 (mc^2*md^2 + restAB[[1]][[1]]^2*mc^2 +
         restAB[[1]][[1]]^2*md^2)]/(2*a*restAB[[1]][[1]]);
  restC = {{Sqrt[ptot^2*a^2 + mc^2]}, {ptot*Cos[phi]*
      Sin[theta]}, {ptot*Sin[phi]*Sin[theta]}, {ptot*Cos[theta]}};
  restD = {{Sqrt[ptot^2*a^2 + mc^2]}, {-ptot*Cos[phi]*
      Sin[theta]}, {-ptot*Sin[phi]*Sin[theta]}, {-ptot*Cos[theta]}};
  ourC = Dot[invTr, restC]; ourD = Dot[invTr, restD];
  Return[{{ourC[[1]][[1]], ourC[[2]][[1]], ourC[[3]][[1]],
     ourC[[4]][[1]]}, {ourD[[1]][[1]], ourD[[2]][[1]], ourD[[3]][[1]],
      ourD[[4]][[1]]}}]]

(*Given the reaction A+B->C+D, this takes the four-momentum of A+B, \
the masses of C and D, and the current spatial constant a, and \
outputs corresponding acceptable results for the four-momentum of C \
and D in in the form {P_C,P_D}. *)

step[vec_, dt_, b_, a1_, a2_] :=
 Table[If[i == 1,
   Sqrt[vec[[1]]^2 +
     a1^2*(vec[[2]]^2 + vec[[3]]^2 + vec[[4]]^2) (a1^2/a2^2 - 1)],
   If[i <= 4, vec[[i]]*a1^2/a2^2,
    Mod[vec[[i]] +
```

```
      dt*(vec[[i - 3]] + vec[[i - 3]]*a1^2/a2^2)/(vec[[1]] +
          Sqrt[vec[[1]]^2 +
            a1^2*(vec[[2]]^2 + vec[[3]]^2 + vec[[4]]^2) (a1^2/a2^2 -
               1)]), b]]], {i, Length[vec]}]


  (*This takes in the current state vector for a \
particle: vec=[E,px,py,pz,x,y,z], the time step taken, the width of \
our toroidal box b, the current universal constant a1, the next value \
of the universal constant a2,and returns an next state vector: \
vec'=[E',px',py',pz',x',y',z'].*)


(*In "CheckCollisions", the input is the input is the particle states \
for baryons M, the particle states for antibaryons Anti, and the \
particle states for photon Phot.  It also takes in the radius of \
interaction for two baryons to interact, or two photons to interact \
to make a baryon rMM.  It has the same for antibaryon-baryon, rMA, \
and for antibaryon-antibaryon, rAA.  It also takes in the current \
value of a, and the masses of baryons (mb) and antibaryons (ma).  It \
outputs a list of all the possible interactions in the form of a \
list, where each element of the list is a possible interaction.  Each \
element of the list is a six-element list \
{{Baryons},{AntiBaryons},{Photons->Barions},{Photons->Barion+\
AntiBaryon},{Photons->AntiBaryons}}, where the first photon-photon \
interaction tells us interactions that can go to baryon-baryon, the \
second tells us interactions that can go to baryon-antibaryon, and \
the third tells us interactions that go to antibaryon-antibaryon.  \
```

```
The elements in each one tells us which address is used for the \
interacting particle. *)


CheckCollisions[M_, Anti_, Phot_, rMM_, rMA_, rAA_, a_, mb_, ma_] :=
 Module[{Int = {}},
  For[i = 1, i <= Length[M], i++,
   For[j = 1, j <= Length[M], j++,
    If[(M[[i]][[-3]] - M[[j]][[-3]])^2 + (M[[i]][[-2]] -
          M[[j]][[-2]])^2 + (M[[i]][[-1]] - M[[j]][[-1]])^2 <= rMM^2,
     AppendTo[Int, {{i, j}, {0, 0}, {0, 0}, {0, 0}, {0, 0}}]]]];
  For[i = 1, i <= Length[M], i++,
   For[j = 1, j <= Length[Anti], j++,
    If[(M[[i]][[-3]] - Anti[[j]][[-3]])^2 + (M[[i]][[-2]] -
          Anti[[j]][[-2]])^2 + (M[[i]][[-1]] - Anti[[j]][[-1]])^2 <=
      rMA^2, AppendTo[
      Int, {{i, 0}, {j, 0}, {0, 0}, {0, 0}, {0, 0}}]]]];
  For[i = 1, i <= Length[Anti], i++,
   For[j = 1, j <= Length[Anti], j++,
    If[(Anti[[i]][[-3]] - Anti[[j]][[-3]])^2 + (Anti[[i]][[-2]] -
          Anti[[j]][[-2]])^2 + (Anti[[i]][[-1]] -
          Anti[[j]][[-1]])^2 <= rAA^2,
     AppendTo[Int, {{0, 0}, {i, j}, {0, 0}, {0, 0}, {0, 0}}]]]];
  For[i = 1, i <= Length[Phot], i++,
   For[j = 1, j <= Length[Phot], j++,
    If[(Phot[[i]][[-3]] - Phot[[j]][[-3]])^2 + (Phot[[i]][[-2]] -
          Phot[[j]][[-2]])^2 + (Phot[[i]][[-1]] -
```

```
           Phot[[j]][[-1]])^2 <=
       rMM^2 && (Phot[[i]][[1]] + Phot[[j]][[1]])^2 -
        a^2 (Phot[[i]][[2]] + Phot[[j]][[2]])^2 -
        a^2 (Phot[[i]][[3]] + Phot[[j]][[3]])^2 -
        a^2 (Phot[[i]][[4]] + Phot[[j]][[4]])^2 > (mb + mb)^2,
      AppendTo[Int, {{0, 0}, {0, 0}, {i, j}, {0, 0}, {0, 0}}]]];
   If[(Phot[[i]][[-3]] - Phot[[j]][[-3]])^2 + (Phot[[i]][[-2]] -
          Phot[[j]][[-2]])^2 + (Phot[[i]][[-1]] -
          Phot[[j]][[-1]])^2 <=
       rMA^2 && (Phot[[i]][[1]] + Phot[[j]][[1]])^2 -
        a^2 (Phot[[i]][[2]] + Phot[[j]][[2]])^2 -
        a^2 (Phot[[i]][[3]] + Phot[[j]][[3]])^2 -
        a^2 (Phot[[i]][[4]] + Phot[[j]][[4]])^2 > (mb + ma)^2,
     AppendTo[Int, {{0, 0}, {0, 0}, {0, 0}, {i, j}, {0, 0}}]];
   If[(Phot[[i]][[-3]] - Phot[[j]][[-3]])^2 + (Phot[[i]][[-2]] -
          Phot[[j]][[-2]])^2 + (Phot[[i]][[-1]] -
          Phot[[j]][[-1]])^2 <=
       rAA^2 && (Phot[[i]][[1]] + Phot[[j]][[1]])^2 -
        a^2 (Phot[[i]][[2]] + Phot[[j]][[2]])^2 -
        a^2 (Phot[[i]][[3]] + Phot[[j]][[3]])^2 -
        a^2 (Phot[[i]][[4]] + Phot[[j]][[4]])^2 > (ma + ma)^2,
     AppendTo[Int, {{0, 0}, {0, 0}, {0, 0}, {0, 0}, {i, j}}]]];
  Return[Int]]

(*ReduceCollisions takes the list of connection given by \
CheckCollisions, and reduces it to a random set of interactions where \
```

```
no particle interacts twice.  It also takes the input "New" as an \
empty vector {}, but it will take in other values for the sake of \
recursion. *)

ReduceCollisions[Int_, New_] :=
 If[Int == {}, Return[New],
  Module[{d = Random[Integer, {1, Length[Int]}]},
   ReduceCollisions[Elim[Int, Int[[d]], 1], Append[New, Int[[d]]]]]]

(*Elimtakes in a list of the collisions, and an element of that list \
that represents a reaction.  It eliminates from the list any elements \
in the list that involve particles that are included in the \
interaction. It also includes the element in the list that it is at.*)

Elim[Int_, i_, N_] :=
 If[N > Length[Int], Return[Int],
  If[in[i, Int[[N]]], Elim[Drop[Int, {N}], i, N], Elim[Int, i, N + 1]]]

(*in returns False if none of the reactants in i appear in v.  It \
returns True if they do *)

in[i_, v_] :=
 If[MemberQ[{{0}},
   Intersection[Join[i[[3]], i[[4]], i[[5]]],
    Join[v[[3]], v[[4]], v[[5]]]]],
  If[MemberQ[{{}, {0}}, Intersection[i[[1]], v[[1]]]],
```

```
   If[MemberQ[{{}, {0}}, Intersection[i[[2]], v[[2]]]], Return[False],
     Return[True]], Return[True]], Return[True]]

(*Collide is not yet finished.  It takes in the states of all the \
particles the radii of interaction, the masses of the baryon mb and \
antibaryon ma, and the current value of a.  It should output the new \
states of the particles once all the collisions have happened.  As of \
now, all it does is remove the particles that have collided.  It \
still has to add the new particles that were created from teh \
collisions. *)

Collide[M_, Anti_, Phot_, rMM_, rMA_, rAA_, a_, mb_, ma_] :=
 Module[{ss =
    ReduceCollisions[
     CheckCollisions[M, Anti, Phot, rMM, rMA, rAA, a, mb, ma], {}],
   Rbaryons = M, Rantibaryons = Anti, Rphotons = Phot, newB = {},
   newA = {}, newPh = {}}, remove = order[ss];
  Do[Drop[Rbaryons, {remove[[1]][[n]]}], {n, Length[remove[[1]]]}];
  Do[Drop[Rantibaryons, {remove[[2]][[n]]}], {n,
    Length[remove[[2]]]}];
  Do[Drop[Rphotons, {remove[[3]][[n]]}], {n, Length[remove[[3]]]};]

(*Given the list of collisions and the lists of baryons M, \
antibaryons Anti, and Photons Phot, I want to create a three element \
list {{M},{Anti},{Phot}} where each element has a list of the \
indieces which will be removed in order.  order does that to the list \
```

```
of collisions. *)


order[clsn_] :=
 Module[{d = {{}, {}, {}, {}, {}}},
  Do[Do[If[clsn[[n]][[m]][[1]] != 0,
     AppendTo[d[[m]], clsn[[n]][[m]][[1]]];
     If[clsn[[n]][[m]][[2]] != 0,
      AppendTo[d[[m]], clsn[[n]][[m]][[2]]]]], {m, 5}], {n,
    Length[clsn]}];
  Return[{Sort[d[[1]], Greater], Sort[d[[2]], Greater],
    Sort[Join[d[[3]], d[[4]], d[[5]], Greater]]}]]
```

References

[1] T. Moore, *A General Relativity Workbook*, 2009, pages 287-289.

[2] V. Mukhanov, *Physical Foundations of Cosmology*, Cambridge University Press, 2005, page 211.

[3] D. Schroeder, *An Introduction to Thermal Physics*, Addison Wesley Longman, United States, 2000, page 299.

[4] A. D. Sakharov, "Violation of CP Symmetry, C-Asymmetry and Baryon Asummetry of the Universe", translation in JETP Lett. 5: 24-27 (1967).

[5] D. Schroeder, *An Introduction to Thermal Physics*, Addison Wesley Longman, United States, 2000, page 292.