High-resolution simulations of cosmic-string evolution. I. Network evolution

David P. Bennett
Physics Department, Princeton University, Princeton, New Jersey 08544
and Princeton University Observatory, Princeton, New Jersey 08540

François R. Bouchet
Institut d'Astrophysique de Paris, 98 bis Boulevard Arago, 75014 Paris, France

(Received 4 December 1989)

We unveil a new cosmic-string evolution code which evolves the string very accurately down to the very small scales (quite close to the separation of our grid points). We argue that high resolution is necessary in order to understand even the large-scale evolution of the string network and we present the results of a careful and comprehensive study of the large-scale evolution of the string network. This comprehensive study includes a large number of numerical tests which guarantees that our results are indeed physical. Finally, our numerical results are compared to an analytic model introduced by Kibble.

I. INTRODUCTION

One of the most exciting predictions of grand unified theories (GUT's) is the possibility that a phase transition at the GUT scale ($\sim 10^{16}$ GeV) could have produced topological defects that would still survive today. Of the three types of defects that could have been produced in this phase transition only cosmic strings seem likely to be present in a detectable abundance.[1,2]

Cosmic strings are also of interest to cosmologists because of the possibility that GUT-scale strings could be responsible for galaxy formation[3] either as seeds for gravitational accretion[4] or, if the strings are superconducting,[5,6] as the energy source for large explosions. The cosmic-string theory of galaxy formation is particularly attractive because cosmic strings have been predicted by theories that were constructed purely for particle-physics reasons. Furthermore, the value of the string's mass per unit length predicted from galaxy-formation considerations $\mu \sim 10^{-6}\ c^2/G$ is in the range to be expected from grand unified theories. This value for μ is also interesting from an observational standpoint because strings with $\mu \sim 10^{-6}\ c^2/G$ will produce a characteristic anisotropy pattern for the cosmic-microwave-background radiation that should soon be detectable.[7] Strings may also be detectable through gravitational lensing,[8,9] or they may be ruled out by the limits placed on low-frequency gravitational radiation from millisecond pulsar timing.[10-14]

Although the cosmic-string galaxy-formation scenario is (in principle) highly predictive, the predictions of the scenario are presently very uncertain. The reason for this is that the details of how cosmic strings evolve in an expanding universe have been poorly understood. We believe that our current simulations have achieved the resolution required to make reasonably accurate estimates of the predictions of the cosmic-string scenario.

The formation of cosmic strings is thought to be governed by the "Kibble mechanism[1]" which states that when the GUT-scale phase transition completes, the Higgs field responsible for the phase transition has a correlation length which must be smaller than the horizon size. On scales larger than the correlation length, the orientation of the Higgs fields are random. The separation of the topological defects formed in the transition should be roughly equal to the correlation length. Vachaspati and Vilenkin[15] have shown that most of the string formed by this "Kibble mechanism" would be part of strings that are infinitely long (or much longer than the horizon). If all the string had been in the form of small loops, then the strings could all oscillate and decay into gravitational radiation leaving no trace of their presence. Infinite strings, however, cannot decay away, so some of these infinite strings are expected to be present today.

Immediately after the strings are formed, their motion is damped by friction with the surrounding matter,[16] but after the Universe cools to $G^{3/2}\mu^2 \sim 10^{12}$ GeV or so, friction becomes negligible, and the strings move freely according to the equations of motion generated by the Nambu action.[17] The basic concept for a discussion of cosmic-string evolution when friction is no longer important is the *scaling solution.*[18-20] In a scaling solution, there are a fixed number (up to statistical fluctuations) of long strings crossing each horizon volume. This implies that the long-string energy density must scale as a^{-4} in the radiation-dominated era and a^{-3} in the matter era where a is the scale factor of a Friedmann-Robertson-Walker universe, so that the strings contribute a (small) fixed fraction of the total energy content of the Universe. If we ignore the interactions that may occur when long strings cross each other, then it is not hard to show[19] that the energy density of the long strings must decrease *slower* than a^{-3}. If string interactions had no influence on the scaling of the long-string energy density with the expansion, then the Universe would eventually become dominated by cosmic strings. For $G\mu/c^2 \sim 10^{-6}$, this would happen very early, contrary to observation.

Therefore, if GUT-scale strings are to exist today, it must be that interactions play an important role in the evolution of long strings. Interactions would cause two string segments to "intercommute" (split and reconnect the other way) when they cross rather than just pass through each other. The question of whether intercommutation occurs depends (in principle) on the details of the field theory that has given rise to the strings, but there is now accumulating numerical evidence that it always occurs[21-23] (for all relative velocities and crossing angles in all theories checked). Interommutation allows the long-string network to lose some of its energy by chopping off loops which can then decay by radiating gravitationally.[24] Since a loop's gravitational decay lifetime is proportional to its length, a scaling solution distribution of loops would look the same at all times when measured in units of the horizon. In order to achieve a scaling solution in the radiation era, we must also require that the net rate of loop production be sufficient to allow the long-string energy density to fall as a^{-4}. This turns out to be quite difficult to verify, and as of yet the only verification has been by numerical calculation such as the one presented herein.

There have been several attempts to tackle this problem analytically which have shed some light on the problem. Kibble[18] has modeled the cosmic-string network with a single parameter: the scale length of the network, L. (Frequent intercommutations imply that both the mean separation between long-string segments and the persistence length along the long strings should be $\sim L$.) The production of loops was described by an unknown loop-production function. Kibble derived a constraint on this loop-production function that must be met in order for a scaling solution to exist, and he showed that the scaling solution (if it exists) is a stable fixed point of his equations. This leaves only two possibilities: either the string network settles down to a stable scaling solution, or the string density grows until the Universe is no longer radiation dominated. Bennett[25,26] studied the Kibble model in great detail and found that a scaling solution is unlikely to exist unless the typical size of the stable non-self-intersecting loops produced by the string network is considerably smaller than the scale length L of the long-string network. If the majority of the string length chopped off the long-string network ends up in loops of size L which do not self-intersect and split up into smaller loops, then the inverse process of loops reconnecting to the long strings reduces the net loop production enough to prevent scaling. On the other hand, if the loops breaking off the long strings are very small or fragment a great deal, then Bennett has shown that a scaling solution should exist.

Another type of analytic approach has been attempted by Mitchell and Turok[27] who studied the statistical mechanics of strings in flat spacetime. They found that the equilibrium distribution of strings in flat spacetime is dominated by the smallest strings they allowed. This suggests that the strings in an expanding universe might show a tendency to chop themselves up into very small pieces which would make a scaling solution inevitable. This seems to be quite similar to what we see in our simulations,[28-30] but unfortunately, no one has been able to come up with a convincing connection between these flat-spacetime statistical mechanics calculations and string evolution in an expanding universe, despite some claims to the contrary.[31] In fact, it is even possible to come up with a loop-production function for the Kibble model[26] which seems to be consistent with the statistical mechanics results *and* cannot produce a scaling solution. Thus, although a lot has been learned from these analytic approaches, they have not been able to tell whether or not a scaling solution exists.

The basic strategy for studying cosmic-string evolution by numerical simulation was developed by Albrecht and Turok,[32] but they were unable to convincingly establish that a scaling solution exists.[26] The first strong evidence for the existence of a scaling solution was provided by our earlier simulations.[33,34] We found that the long-string density at scaling was more than an order of magnitude larger than the "one long string per horizon" value that had been expected[18-20] and that the size of the loops produced was more than two orders of magnitude smaller than the horizon size. In fact, loop production at small sizes was so important that our early results for the long-string density were sensitive to our lower cutoff on loop size. But by running with a high density of sampling points along the string and with small cutoffs on loop size, we were able to achieve good enough resolution so that the behavior of the long strings no longer had much dependence on the small loop cutoff. However, the loop distribution remained cutoff dependent, and since many of the observational consequences of cosmic string depend sensitively on the loop distribution, we decided that a major improvement in our code was called for.

The main difficulty encountered in evolving cosmic strings numerically is the presence of large numbers of "kinks" which are physical discontinuities in the velocity of and the tangent vector to the string. Four kinks (one left moving and one right moving on each piece of string) are formed whenever two pieces of string intercommute, and they remain infinitely sharp at the resolution of the simulations. As we show in Appendix B, their amplitude decays very slowly ($\sim a^{-0.14}$ in the radiation era) due to the expansion. So, the kinks are responsible for lots of small-scale structure on the long strings, and it is this small-scale structure which is responsible for the fact that the only non-self-intersecting loops produced tend to be very small. Thus, in order to have an accurate cosmic-string-evolution code, we must go to great lengths to see that the kinks are preserved. This is exactly what we have done with our new code (hereafter code II). As we shall see, the results of this code II are actually quite similar to our highest-resolution results with the old code (hereafter code I). This gives us confidence that both of our numerical codes are quite reliable, and it has allowed us to substantially reduce our estimated errors.

The main results of our simulations of string evolution have been summarized in Ref. 28. We have found that in the radiation era, the evolution of the long-string network is well described by a simplified version of Kibble's one scale model. (This is the same model used by Albrecht and Turok[31] in their recent paper describing results from an improved version of their numerical simulation code.) However, we also find[28-30] that about 45% of the string energy is in modes with a wavelength smaller than the persistence length of the network $\xi \sim L$. This energy is smoothly distributed in all wavelengths from ξ down to the limit of our resolution. Since our resolution (as measured in horizon units) increases with time, this distribution is time dependent. Because the total energy in short wavelengths is constant, this time dependence has little effect on the long-string density, but it does have a large influence on the distribution of stable non-self-intersecting loops produced by the long strings. We find that as our simulations run and more and more kinks bulk up on the long strings, the stable loops that are produced are smaller and smaller. Even when we introduce a large density of kinks into our initial conditions, we do not produce a scaling distribution of loops.

This has rather important consequences for the upper bound that can be placed on $G\mu$ from millisecond pulsar timing measurements.[10,13] The lower resolution simulations of Albrecht and Turok[31] (AT) *have* indicated an apparent scaling distribution of loops, but as we show in Appendix A, the smoothing length in their simulations scales with the horizon size and is only slightly smaller than the scale length of the long strings. Thus, their scaling distribution of loops is likely due to the scaling of their smoothing length. If their distribution of loops was real, it would indicate a limit[31,35] of $G\mu < 2.5\times 10^{-8}$ which would rule out the cosmic-string galaxy-formation scenario. (We have inserted Taylor's latest number[13] into the formulas of Refs. 31 and 35, and we have set $c \equiv 1$.) However, when the evolution of the loop-production function as seen in our high-resolution simulations is properly taken into account,[14] the best limit that can be obtained is $G\mu < 4\times 10^{-6}$ which is only slightly better than the best previous limit, $G\mu < 5\times 10^{-6}$, which comes from upper bounds on the anisotropy of the microwave background.[7]

We also differ somewhat with the recent results of AT in our values for the long-string density. We find that $\rho_{\text{LS}} = (52\pm 10)\mu/H^2$ in the radiation era and $\rho_{\text{LS}} = (31\pm 7)\mu/H^2$ in the matter era where $H = a\tau = a\int dt/a$ is the horizon length (and *not* the Hubble constant). AT (Ref. 31) have found $\rho_{\text{LS}} = 210\mu/H^2$ in the radiation era and $\rho_{\text{LS}} = 64\mu/H^2$ in the matter era. In the main body of their paper, they quote their errors as ±40% or 50%, but they admit to not systematically investigating errors associated with their solution. They also suggest in a "note added in proof" that a comparison with unpublished results of their flat-spacetime simulations indicates that their value for ρ_{LS} could be too high by a factor of 4. More recently, Allen and Shellard[36] (AS) have developed a new string simulation code and found that $\rho_{\text{LS}} = (64\pm 16)\mu/H^2$ in the radiation era in confirmation of our results. They also confirm qualitatively most of our results on the small-scale structure of the string network and the nature of the loop production function.

In the next section, we will give a detailed explanation of the numerical methods of both of our codes. We will compare these methods with each other as well as with the codes of AT and AS. We will also show the results of some simple tests of our codes. In Sec. III, we will display our main results on the evolution of the string network in the radiation, matter, and transition eras. We will also show the results of many tests that we have performed to ensure that our results are not contaminated by numerical effects. In Sec. IV we will compare our results to a simple version of Kibble's one scale model and to some extensions of this model that were suggested by AT (Ref. 31), and we will make some concluding remarks in Sec. V. A discussion of the numerical smoothing in string evolution codes is given in Appendix A, and a derivation of the formula which governs the (very slow) decay of kinks with the expansion is given in Appendix B.

The reader should note that there are several very important issues that have been left out of this paper. The main points of our study of the evolution of the small-scale structure on the long strings and its effects on the loop-production function have been discussed in previous papers[28,29] and will be covered in great detail in the sequel[30] to this paper. The two-point correlation function of cosmic-string loops was studied in Ref. 37.

II. NUMERICAL METHODS

In this section, we briefly review the basic ingredients of the numerical calculations in both the old and new versions of our code. Readers who are not interested in the details of our numerical calculations might prefer to skip all but the latter part (Sec. II E) of this fairly technical section. We shall also point out some of the important differences between each of our codes and the codes of AT (Ref. 31) and AS (Ref. 36). The basic strategy of our codes (as well as those of AT and AS) is to generate some "reasonable" initial conditions, and to evolve the resulting network forward in time. The time evolution involves solving the equation of motion, detecting the string crossings and then performing the intercommutations. Each of these steps must be done every time step.

A. Generation of the initial conditions

The numerical approach taken here is first to generate "reasonable" initial conditions, i.e., to create a string configuration which is a network of random walks of given step size ξ_0 in a box with periodic boundary conditions. To do so, we simply follow the procedure introduced by Vachaspati and Vilenkin.[15] (One draws discretized random phases for each cell of a cubic periodic lattice; assuming the phases vary in the minimal way, if the winding number around an edge is nonzero, the link is occupied.) Sampling points and their attached pointers are then laid down accordingly. The initial condition code is fully vectorized, and takes very little time to run.

A common misconception about these initial conditions has arisen from the fact that this algorithm was originally developed to study the phase transition which originally produced the strings. It is frequently assumed that the Vachaspati-Vilenkin-type initial conditions are used as the starting point for string simulations because they are the "correct initial conditions for strings." This argument neglects the fact that soon after string formation the motion of the strings is heavily damped by friction[16] with the surrounding hot gas. One might then expect that the "correct initial conditions" would be the configuration at the end of the heavy damping era, but this is not quite right either.

The point is that the string system is supposed to settle down to a scaling solution shortly after the heavy damping period. Since our main goal is to study the properties of this scaling solution, we would like to start most of our runs as close to the scaling solution as possible. It is also desirable to do some runs which start in configurations which do not resemble the scaling solution in some way in order to test Kibble's prediction that the scaling solution is a stable fixed point of the system.

The Vachaspati-Vilenkin initial conditions have the advantages that they are very easy to implement and that they look like the scaling solution (i.e., a network of random walks) on scales larger than the initial persistence length ξ_0. The one free parameter that the Vachaspati-Vilenkin initial conditions allow is the ratio of ξ_0 to the initial horizon size. Adjusting the initial horizon size is equivalent to adjusting the density of strings in a horizon volume, so we can look for the relaxation to the scaling solution from high- and low-density initial conditions simply by adjusting the initial horizon size.

In code I, which (as we shall see) did not deal with kinks as well as code II does, we decided to roundoff the sharp corners produced by the Vachaspati-Vilenkin procedure in order to diminish the number of discontinuous derivatives the evolution program had to cope with. We also added some small transverse velocities with a wavelength of ξ_0 in order to avoid degeneracies that occur when stationary loops collapse.[38]

Code II has no difficulty with kinks in the initial conditions because kinks are evolved just as accurately as the rest of the string with code II. As we shall see, one of the main differences between the strings in these smooth initial conditions and the strings at the end of a run is that at the end of a run the strings have gained a lot of small-scale structure by the end of the run. Therefore, the way to get closer to the scaling solution at the start is to add lots of structure on the smallest scales. We have accomplished this by including an option to add random velocities of fixed amplitude at each and every point along the string. As we shall see below, the velocity component parallel to the direction of the string is unphysical and is constrained to be zero by our gauge condition. Therefore, we simply throw away the parallel component of the velocity. We shall see that starting a run with "kinky" initial conditions does indeed bring us closer to the scaling solution.

For comparison, AT use the Vachaspati-Vilenkin initial conditions without modification while AS do the same sort of rounding that we do. AS also add velocities of $1/\sqrt{2}$ to the strings with a coherence length of about ξ_0. These differences seem to have little effect on the final results. Another difference between the AS code and ours is that they have randomly reassigned the directions of all the string in the box. This means that their string networks can have nonzero winding numbers. (The Vachaspati-Vilenkin procedure ensures that all the winding numbers are zero.) This should have noticeable effects only after a simulation has been run so long that the effects of the finite box size and the periodic boundary conditions become important.

B. Numerical solution of the equation of motion

1. Equations of motion

The equations of motion for cosmic strings can be derived from the Lagrangian of the field theory that produces the strings, but the Nambu action[17] that describes strings of zero thickness is a very good approximation for cosmic strings.[39,40] The equations of motion for strings are most conveniently written in comoving coordinates where the Friedmann-Robertson-Walker metric takes the form $ds^2=a^2(-d\tau^2+d\mathbf{r}^2)$. The comoving spacial coordinates of the string, $\mathbf{x}(\tau,\sigma)$, are written as a function of conformal time τ and the length parameter σ. It is also convenient to choose a gauge in which the unphysical parallel components of the velocity vanish, i.e.,

$$\dot{\mathbf{x}}\cdot\mathbf{x}'=0 \; . \tag{2.1}$$

In these coordinates, the Nambu action yields the equation[19]

$$\ddot{\mathbf{x}}+2\left[\frac{\dot{a}}{a}\right]\dot{\mathbf{x}}(1-\dot{\mathbf{x}}^2)=\left[\frac{1}{\epsilon}\right]\left[\frac{\mathbf{x}'}{\epsilon}\right]' \; . \tag{2.2}$$

Overdots denote derivatives with respect to conformal time τ; primes denote partial derivatives with respect to σ. $\epsilon\equiv\sqrt{\mathbf{x}'^2/(1-\dot{\mathbf{x}}^2)}$ is the string energy per unity σ (in comoving units), so that $\mu a\int\epsilon d\sigma$ is the string energy. Equation (2.2) implies that

$$\frac{\dot{\epsilon}}{\epsilon}=-2\frac{\dot{a}}{a}\dot{\mathbf{x}}^2 \; . \tag{2.3}$$

The only gauge freedom still remaining is the choice of the parametrization by σ at a particular time. The choice usually made is to choose σ such that $\epsilon=1$ in the initial conditions.

Some very important insights into the problem of string evolution can be obtained by looking at the flat-space limit of Eq. (2.2):

$$\ddot{\mathbf{x}}=\mathbf{x}'' \; , \tag{2.4}$$

where we have set $\epsilon=1$. With the gauge condition (2.1), (2.4) implies that $\dot{\mathbf{x}}^2+\mathbf{x}'^2=1$. The general solution to (2.4) is

$$\mathbf{x}=\tfrac{1}{2}[\mathbf{a}(\sigma-\tau)+\mathbf{b}(\sigma+\tau)] \; , \tag{2.5}$$

where

$$\mathbf{a}'^2=\mathbf{b}'^2=1 \; . \tag{2.6}$$

We require the functions $\mathbf{a}(\sigma-\tau)$ and $\mathbf{b}(\sigma+\tau)$ to be continuous (because the string must be continuous), but their derivatives need not be. In fact, whenever two strings intercommute, both $\dot{\mathbf{x}}$ and $\mathbf{x}'$ will be discontinuous on both new string segments at the point of intersection. From the general solution (2.5), it is clear that these initial discontinuities are composed of right- and left-moving pieces (known as kinks) which will travel down the string at the speed of light. In flat space if we neglect gravitational radiation back reaction, it is clear that these kinks will remain infinitely sharp and retain their amplitudes indefinitely.

2. *Evolution in code I*

In this subsection, we describe the method used to solve the equation of motion (2.2) in our original code.

The strings are represented by a series of "sampling" points which are linked by "upstring" and "downstring" pointers so that neighboring points can be determined. The string positions between the points are obtained by linear interpolation between the sampling points (a higher-order interpolation scheme would be less accurate, since the strings are not very smooth.) In order to evolve the positions and velocities of the sampling points, we discretize the equation of motion (2.2). After many tests involving many different schemes, we decided to use a simple modified leapfrog scheme with spatial derivatives at midpoints obtained by finite differences. Our discretized version of (2.2) is

$$\mathbf{v}(\tau+\Delta\tau/2)=\mathbf{v}(\tau-\Delta\tau/2)+\Delta\tau\left[-2h(\tau)\mathbf{v}(\tau)(1-\mathbf{v}^2)+\frac{2}{\epsilon(\sigma+\Delta\sigma/2)+\epsilon(\sigma-\Delta\sigma/2)}\frac{1}{\Delta\sigma}\times\left[\frac{\mathbf{x}(\sigma+\Delta\sigma)-\mathbf{x}(\sigma)}{\epsilon(\sigma+\Delta\sigma/2)\Delta\sigma}-\frac{\mathbf{x}(\sigma)-\mathbf{x}(\sigma-\Delta\sigma)}{\epsilon(\sigma+\Delta\sigma/2)\Delta\sigma}\right]\right], \tag{2.7}$$

$$\mathbf{x}(\tau+\Delta\tau)=\mathbf{x}(\tau)+\Delta\tau\mathbf{v}(\tau+\Delta\tau/2), \tag{2.8}$$

where $\mathbf{v}=\dot{\mathbf{x}}$ is the velocity which is stored at half-integral time steps, and $h=\dot{a}/a$ is the Hubble "constant." The local "energy density" ϵ is stored at integral time steps but half-integral spacial steps. In order that the scheme be (formally) second-order accurate in time, $\mathbf{v}$ must be determined (to first order) at time τ for use on the right-hand side of (2.7). This is done by using the first-order version of (2.7), i.e., replacing $\mathbf{v}(\tau)$ with $\mathbf{v}(\tau-\Delta\tau/2)$, and $\mathbf{v}(\tau+\Delta\tau/2)$ with $\mathbf{v}(\tau)$ in (2.7), and taking a half-time step. We also independently evolve (with a semi-implicit scheme) the local "energy density" ϵ according to Eq. (2.3), which yields

$$\epsilon(\tau+\Delta\tau)=\epsilon(\tau)\frac{1-A}{1+A},\quad A=\Delta\tau\left[\frac{\dot{a}}{a}\right]\mathbf{v}^2. \tag{2.9}$$

($\mathbf{v}^2$ is obtained by averaging over the end points.) The accuracy of the calculation can then be checked by comparing the evolved value of ϵ with the value computed from $\dot{\mathbf{x}}$ and $\mathbf{x}'$ (see below). Each loop carries its own time step satisfying the Courant stability condition $[\Delta\tau<\min(\epsilon)\Delta\sigma]$, and our time-step halving routine preserves the second-order accuracy of the overall calculation. These routines are fully vectorized.

The major difficulty encountered by this method is due to the constant production of kinks whenever strings cross and intercommute. As we shall see below, these kinks remain infinitely sharp (since we ignore gravitational radiation and its back reaction) and their amplitude changes very slowly with time. (See Appendix B for a proof of this.) For these reasons we have made a considerable effort to evolve these kinks properly in our simulations. In a code without numerical diffusion these discontinuities would cause some rather serious problems since short-wavelength modes are handled very poorly by finite differencing schemes. (One can analyze this by considering that a discretized operator, such as a spatial derivative, is equal to the continuous one times a transfer function which depends on the wavelength. This transfer function generically behaves as some power of $\sin(z)/z$, with $z=2\pi\epsilon\Delta\sigma/\lambda$; i.e., the discretized operator is not a very good approximation to the continuous operator for waves with wavelengths close to the Nyquist wavelength.) One effect of these large errors for the short-wavelength modes is that these modes will propagate at different velocities. This has a rather dramatic effect on a sharp discontinuity such as a kink: as the kink propagates down the string, these erroneous velocities for the short-wavelength modes cause the short-wavelength modes to decorrelate. This causes the kinks to spread out somewhat, but more importantly, it also causes short-wavelength oscillations to grow in sections of string that should be straight (see Fig. 4 below for a graphical illustration of this).

To avoid this problem, we introduce some numerical diffusion only *when and where* these short-wavelength instabilities start to develop. This is accomplished by checking the quantity

$$\mathbf{v}^2+\left[\frac{1}{\epsilon}\mathbf{x}'\right]^2, \tag{2.10}$$

which is conserved by the equations of motion and should remain equal to unity. This check is possible because we evolve ϵ separately even though it is not independent of $\mathbf{x}$ and $\mathbf{v}$. By averaging the velocities over neighboring points whenever this quantity gets to be larger than 1 by a few percent, we are able to smooth out the short-wavelength instabilities (see Fig. 4). This also increases the width of the kinks somewhat, but since the numerical diffusion is invoked *only* when an instability starts to develop, the kinks should be smoothed out much less than in routines which smooth the kinks at every time step such as the codes of AT (Refs. 31 and 32).

One apparent drawback of our numerical diffusion scheme is that it does not preserve the gauge condition (2.1). We have tried imposing this condition numerically, but we have found that this further degrades our accuracy. One might argue that it would be better if we imposed the gauge condition anyway, but we prefer an approximate solution that violates the gauge condition to a *less accurate* approximate solution that preserves it.

A somewhat more serious difficulty with our diffusion scheme is that it systematically underestimates $\mathbf{v}^2$. This means that according to (2.3) [and (2.9)] the value of $\dot{\epsilon}$ will be systematically overestimated causing unphysical stretching of the string. In order to prevent this sort of systematic error from developing, we replace $\mathbf{v}^2$ in (2.9) $[\mathbf{v}^2+1-(\mathbf{x}'/\epsilon)^2]/2$ whenever $\mathbf{v}^2+(\mathbf{x}'/\epsilon)^2<1$. This correction certainly removes most of the systematic error in the evolution of ϵ, but it is difficult to tell whether any remaining systematic error is left. In fact, the possibility that this procedure might undercorrect or overcorrect the error in the evolution of ϵ was one of the main factors contributing to the relatively large error bars quoted in our first paper.[33] As we shall see, our fears were greatly exaggerated; the systematic error left by this procedure seems to be too small to detect.

Overall, we have found that this evolution scheme has worked fairly well. The method of dealing with kinks has been carefully tested on loops with a few kinks where it has been found to work very well. All seemed fine until we began to do large high-resolution runs on the Cray-2 and found that by the end of the run there was one kink for every three sampling points on the string. For just the long-string network the ratio was more like 1:5, but this still seemed worrisome because a typical kink would have spread out over a few points by the end of the run. We were also concerned because the errors seemed to build up quite systematically during a run, so we were not sure quite how long we could run the simulation and still trust the results. These were the main factors which motivated us to write code II, but as we will see when we compare the results of codes I and II, most of these worries were not justified.

3. Evolution in code II

As we mentioned in the Introduction, these kinks are the dominant feature of the strings on scales smaller than L, and the small-scale structure is responsible for determining the distribution of non-self-intersecting loops that are generated by the long-string network. Therefore, it is very important to evolve the kinks accurately and preserve their discontinuous nature. This is quite easy to do in flat space where we can simply use the solution (2.5) of the corresponding equation of motion (2.4). In fact, a very large class of numerical integration schemes [including (2.7) and the AT scheme] can be made exact in flat space *if* the time step is set equal to the space step, i.e., $\Delta\tau=\epsilon\Delta\sigma$. But in the expanding universe, ϵ varies from point to point, and we must have $\Delta\tau\leq\min(\epsilon)\Delta\sigma$ in order to satisfy the Courant stability condition. Therefore, the advantages of the simple flat-spacetime solution can only be realized if our routine is exact in flat space for an *arbitrary* time step (limited perhaps by the Courant condition).

It is not too difficult to construct such a scheme if one uses the appropriate dependent and *independent* variables. If we define

$$\mathbf{p}(\tau,s_-)\equiv\frac{\mathbf{x}'}{\epsilon}-\mathbf{v},\quad \mathbf{q}(\tau,s_+)\equiv\frac{\mathbf{x}'}{\epsilon}+\mathbf{v}\ , \tag{2.11}$$

where $\mathbf{p}^2=\mathbf{q}^2=1$ and

$$s_\pm\equiv\int\epsilon d\sigma\pm\tau\ , \tag{2.12}$$

then the equation of motion (2.2) implies

$$\begin{aligned}\dot{\mathbf{p}}&=-\frac{\dot{a}}{a}[\mathbf{q}-(\mathbf{p}\cdot\mathbf{q})\mathbf{p}]\ ,\\ \dot{\mathbf{q}}&=-\frac{\dot{a}}{a}[\mathbf{p}-(\mathbf{p}\cdot\mathbf{q})\mathbf{q}]\ ,\\ \dot{\epsilon}&=-\frac{\dot{a}}{a}\epsilon(1+\mathbf{p}\cdot\mathbf{q})\ .\end{aligned} \tag{2.13}$$

The crucial change here has been the change to the independent variables $s_\pm$ for $\mathbf{p}$ and $\mathbf{q}$. This change has left only terms proportional to $\dot{a}/a$ on the right-hand side of the equation, so the flat-space equations are just $\dot{\mathbf{p}}=\dot{\mathbf{q}}=\dot{\epsilon}=0$. Clearly a numerical solution to these equations can be made exact in flat space for any time step satisfying the Courant condition.

The method we use to solve Eqs. (2.13) and (2.12) is quite similar to the method of characteristics.[41] In fact, the curves of constant $s_\pm$ are the characteristics of (2.2). However, the method of characteristics would have us solving (2.13) on a grid of the intersections of the characteristic curves. This would be just fine if we only needed the solution to Eq. (2.13), but we also want to check for crossings. Therefore we need the solution to (2.13) on surfaces of constant τ. Figure 1 shows the relationship between the τ-σ grids used and the characteristic curves of constant s_+ and s_-. Different pairs of characteristic curves do not meet at the same values of τ because ϵ varies from point to point. One way around this difficulty would be to interpolate between the characteristic curves to obtain $\mathbf{p}$ and $\mathbf{q}$ on the vertices of τ-σ grid. The great danger with this procedure is that it might

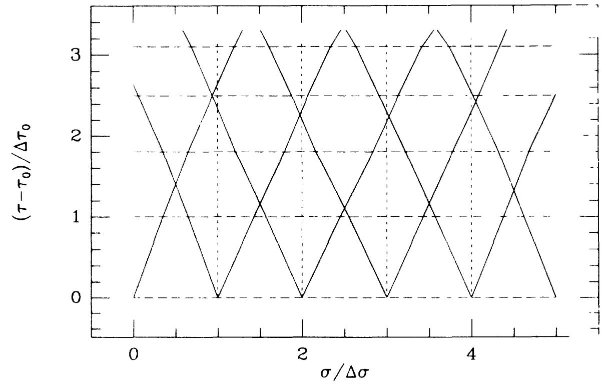

FIG. 1. This figure shows the different lattices used in our code II evolution scheme. The solid lines are the characteristic curves which are surfaces of constant s_- (moving to the right) and constant s_+ (moving to the left) while the long-dashed lines are surfaces of constant σ and the short-dashed lines are surfaces of constant τ which represent consecutive time steps.

smooth out the kinks which appear as discontinuities in **p** and **q** at about 20% of the grid points.

We have avoided this danger with the following strategy: we store **p** and **q** on grids in s_+ and s_- while ϵ and **x** are still stored on a σ lattice. A bit of numerical machinery is required to keep track of how the $s_\pm$ lattices move with respect to the σ lattice. In order to avoid any artificial smoothing of the kinks, we do not interpolate to get values of **p** or **q** between the lattice points. Instead we take **p** and **q** to be constant over a segment on the s_+ and s_- grids. (By a segment we mean just the bit of string between two grid points.) Thus, **p** and **q** are constant between grid points and discontinuous at grid points. These discontinuities can have large amplitudes when a kink is present, but they are small otherwise.

In order to use Eqs. (2.13) to evolve **p** (or **q**), we must first obtain values for $\mathbf{q}(\mathbf{p})$ and $\mathbf{p}\cdot\mathbf{q}$. Since **p** and **q** are taken to be constant over a segment, the proper values for $\mathbf{q}(\mathbf{p})$ and $\mathbf{p}\cdot\mathbf{q}$ are obtained by averaging the discrete values of $\mathbf{q}(\mathbf{p})$ that intersect the area in the τ-σ plane that is traversed by the segment of the $s_-(s_+)$ lattice in the time interval $\Delta\tau$. The right-hand side of the ϵ equation is done in the same way.

To calculate the positions from **p** and **q** one can use either

$$\dot{\mathbf{x}}=\tfrac{1}{2}(\mathbf{q}-\mathbf{p}) \tag{2.14}$$

and integrate from the initial positions, or

$$\mathbf{x}'=\tfrac{1}{2}\epsilon(\mathbf{p}+\mathbf{q}) \tag{2.15}$$

and integrate along σ from a known position. If we limit ourselves to initial values for **p** and **q** which can be represented exactly on our lattices with our "interpolation" procedure, then both methods for calculating the positions are exact in flat space. In practice, we combine both of these methods: at every time step, we move the positions forward in time via (2.14), but after every ten time steps or so, we reset all the positions using (2.15) while keeping the center-of-mass position fixed. This cancels the cumulative numerical errors that might otherwise build up between neighboring points. [Once *all* the bugs were removed from the program though, the cumulative errors canceled by this procedure became almost negligibly small for (typical) runs of 1000 time steps.]

A potential difficulty with the evolution scheme as outlined above is that in some places along the string, the separation of neighboring points on the $s_\pm$ grids can become very small. Intercommutation can cause this to occur very frequently, but it can also occur when a section of one of the $s_\pm$ grids has been on a piece of string that has been moving at a high velocity for a long period of time. (This causes ϵ to decrease quite rapidly.) Since the size of each string's time step must be smaller than half the minimum separation on the $s_\pm$ grids for that string, small separations on the $s_\pm$ grids could slow down the code very significantly. To avoid this, we have a routine which removes points from the $s_\pm$ grids when their separation becomes too small (< one-half of the mean separation on the σ grid). This routine changes three segments on a $s_\pm$ grid to two while conserving both energy and momentum. This is accomplished by interpolating on the $\mathbf{p}^2(s_+)=1$ [or $\mathbf{q}^2(s_-)=1$] sphere to obtain new values of $\mathbf{p}(s_+)$ $[\mathbf{q}(s_-)]$ at the new points on the $s_+(s_-)$ grid. This has the effect of smearing the structure on scales smaller than the mean σ-grid spacing and moving the structure to scales between one and two steps on the σ grid. Since this smoothing routine is not called very frequently (in a typical simulation only one point out of ten will ever be effected), it is fair to say that our resolution scale is quite close to the spacing on the σ grid.

AT have suggested[42,43] that our smoothing routine might *increase* the size of the kinks and the amount of small-scale structure on the strings, but this is not the case. Our smoothing routine clearly moves the structure from scales that are smaller than our mean grid spacing to a scale slightly larger. In fact, our routine does not change the mean value of the parameter θ defined in Ref. 43, so according to Albrecht's arguments, this routine should have no net effect on the loop-production rate.

We have performed numerous numerical tests on this evolution scheme. Some of these will be discussed in Sec. II D. Formally, this new evolution scheme is only first-order accurate, so one might expect that the leapfrog scheme of code I would be more accurate for smooth strings because it is second order accurate. This is not the case. We have tested both schemes against a run of the leapfrog code with 40 times the number of points, and found that the new scheme is more accurate than the old one even for loops without kinks. The reason for this is that the errors in code II are all proportional to $\dot{a}/a$ which is quite small compared to the frequency of the dominant modes on the string. In short, this evolution scheme has the advantage that all the violent oscillation of the strings has been removed by the choice of independent variables. The evolution that we are left with is just the very slow variation of **p** and **q** due to the expansion.

AT's new code uses an equation that is quite similar to Eq. (2.13), but they use τ and σ as their independent variables. Because ϵ varies along the string and their time step is limited to be $<\min(\epsilon)\Delta\sigma$, they cannot take time steps that are large enough so that the characteristics that start at integral multiples of $\Delta\sigma$ will meet at integral multiples of $\Delta\sigma$ at the end of the time step. (See Fig. 2.) Instead of keeping the values of **p** and **q** at fixed values of $s_\pm$ like we do, they interpolate at every time step to get new values for **p** and **q** at integral multiples of $\Delta\sigma$. As we demonstrate in Appendix A, this incessant interpolation smooths out of the short- and medium-wavelength structure considerably, and it seems clear that this smoothing contributes substantially to the differences in our results. Despite their pervasive smoothing, AT have suggested[31] that their code might somehow "approximate the continuum limit" better than our code. It is difficult to imagine how this could possibly be the case since our smoothing procedure is quite similar to theirs. The main difference is that our smoothing routine is invoked about 10^4 times in a typical run with 10^5 points and 1000 time steps while AT would invoke smoothing 10^8 times in such a run.

Like us, AS have gone to some effort to come up with a code which will preserve the kinks. They have chosen a

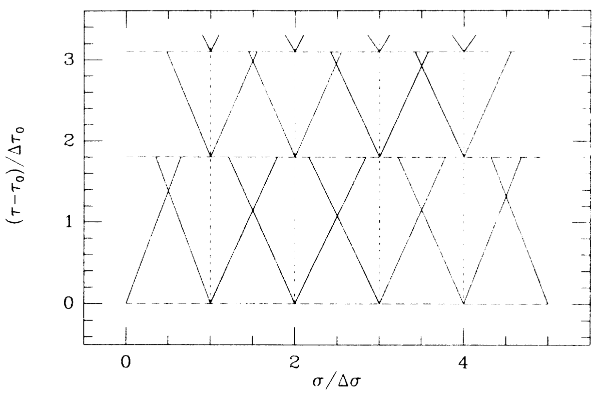

FIG. 2. This figure shows how the lattice used in the AT evolution scheme differs from our lattice (Fig. 1). The main difference is that they require the s_+ and s_- grids (solid lines) to be lined up with the σ grid (short-dashed lines) at every time step. This implies that they must reparametrize their $s_\pm$ grids at every time step.

"total variation nonincreasing" scheme that was originally developed to handle shock waves. They claim that their routine halts the spreading of kinks after they have reached a width of 3 or 4 grid points. This is roughly comparable to the amount of spreading we observed in code I (although this spreading is never "stopped" in code I), but it is considerably more spreading than we have in code II in which kinks are only very rarely spread over a distance as large as two grid spacings. This means that code II can be run with a factor of roughly 3 or 4 fewer grid points (on each grid) than code I or the AS code in order to achieve the same accuracy. Thus, although code II is considerably slower per step per grid point than code I or the AS code, it is much faster than code I and may well be faster than the AS code when run at the same level of accuracy.

C. Crossing detection and intercommutation

Our basic crossing detection routine has remained unchanged in our new code, but we have added some new features in order to take advantage of the improvement in resolution allowed by our new string representation and differential equation solver. The basic routines are fully vectorized, but in code I they were the most time-consuming part of the program. In code II crossing detection and the differential equation solver take comparable amounts of time. In this section, we will first discuss the basic routines which are common to both of our codes, and then mention the refinements we have made in code II.

1. Our basic crossing detection scheme

In order to obtain a list of candidate segments that might have crossed, we first lay down a detection grid dividing the simulation box in small cubes whose optimal size is determined by the maximum distance separating segments that might have crossed during a time step. We then fill a linked list of all our segments with the heads of list being the detection array. For each segment a list of neighbors eligible for crossing is built by using the linked lists corresponding to the cell containing the segment and its 26 nearest neighbors. (In fact, the search may be reduced to seven neighboring cells most of the time.) This list of candidates is scanned to throw out all pairs of segments that are too far away to cross.

A detailed checking for crossing is done for the remaining pairs of string segments. First, the volume of the tetrahedron spanned by the four points on the two segments is checked to see whether it changed sign during the time step. If it did, the configuration is checked at the time the volume is zero to see if a crossing did really occur (the positions of the points are extrapolated linearly between time steps). The crossing time obtains by solving a cubic equation which is (usually) done using Newton's method, and the crossing condition may then be expressed as sign checks of expressions involving scalar and vector products of the edge vectors at crossing. This overall procedure is exact but for the tetrahedron sign check since there might be a crossing with a double sign change. We tested for this by turning off this screening test in one of our runs and we found that we actually missed only a few crossings corresponding to very degenerate loops. So, we decided to keep the test in order to speed up the calculation.

Finally, the internal dynamics of loops is on a smaller time scale than the displacement of complete loops. A given loop is thus checked for self-crossings at each of its individual time steps, while the crossings between loops are done only at each system time step when all loops are synchronized. This in particular allows us to "focus" the detection array on each loop checked for self-intersection.

2. Intercommutation with code I

We keep track of the neighboring points along a string by means of forward and backward pointers, so the interchange of partners itself is straightforward. Before performing the intercommutations, we implement a minimum cutoff on the size of the loops that can be formed by requiring that a new loop have a minimum number of points. This allows us to vary the lower cutoff to analyze its effect on the physical quantities that we are attempting to measure. To do this we simply perform the crossings chronologically and inhibit further crossings for string segments in the vicinity of an interchange. We also must do some initial smoothing of the kinks in order to minimize the amount of numerical diffusion that will be invoked by the evolution routine. This is accomplished by averaging the positions and velocities of the points on either side of the kinks with their nearest neighbors. At this stage, we also update a "genealogical tree," which records in a compact form, for each crossing, the labels of the "parent" loops and of the "child" loops, as well as such relevant quantities as "birth" and "death" times, energies, center-of-mass position and velocity, etc. This enables us to get *a posteriori* a detailed picture of the string system evolution.

3. *Refinements for code II*

In code II the crossing detection scheme outlined above is only the first step of the crossing detection procedure. Once candidates are selected with the routine described above, they are checked again for crossing with a different sort of interpolation between neighboring points. As described above, the exact string positions are obtained from Eq. (2.15) using the fact that **p** and **q** are constant on segments of their lattices and discontinuous at $s_\pm$ grid points. It is then straightforward to calculate the "exact" string crossing point if we neglect the time evolution of **p** and **q** due to the expansion for a fraction of a time step. Once we have the crossing point, we interchange partners, and we add one new right-moving and left-moving grid point at the position of the new kinks on each of the new string segments. This intercommutation procedure is exact in flat space, and so the only error in the expanding universe case is that we do not include the evolution of **p** and **q** over the fraction of a time step between the end of the time step and the time when the crossing occurs.

There is also a potential problem with adding new points on the $s_\pm$ grids for the new kinks generated at every intercommutation. Occasionally, a new kink will form very close to an old $s_\pm$ grid point, and this can make the minimum separation between grid points extremely small. This would force the time step to also be very small via the Courant condition and would virtually halt the code. To avoid this, we must invoke the routine described in the previous section to delete points from the $s_\pm$ grids if their separation is too small (roughly less than half the mean separation on the σ grid). This routine is used in such a way as to avoid altering the segments containing the new kinks in order to avoid unnecessary smoothing of the kinks.

The minimum loop size that we allow in our simulations has generally been set to three points (that is, three points on the σ grid; these loops typically have from two to four points on their $s_\pm$ grids). We cannot set this cutoff to be smaller because our intercommutation routine cannot handle the reconnection of loops smaller than this. With this finite cutoff, we have found that there are a substantial amount of energy in loops which are not much larger than the cutoff. It is quite natural to wonder if these loops would fragment into even smaller loops if they were not prevented by the finite lower cutoff. Therefore, we have added a small loop fragmentation routine in order to check this and to further increase our resolution.

This small loop fragmentation routine is called whenever a loop is produced with fewer than about 15–20 points. The fragmentation routine then finds the self-crossings of the loop and performs the fragmentation at the first crossing. These small loops are assumed to evolve as though they were in flat space. Each new loop that is produced in this fragmentation process is checked for crossings until only non-self-intersecting loops are left. Throughout this process new grid points are added on the $s_\pm$ grids whenever new kinks are created, and we do not delete any grid points until the fragmentation process is completed. When the fragmentation process is done, the resulting stable loops which are not too small (i.e., those that are as large as the smallest loops we would have allowed before we implemented this procedure) have all the very small segments of their $s_\pm$ grids deleted and are returned to the simulation. The smaller loops are never checked for crossings again.

4. *Comparison with AS and AT*

The crossing detection scheme used by AS is essentially the same as our basic crossing detection scheme described above. This procedure is essentially exact up to details of the interpolation between points. The AT crossing detection scheme is too complicated to describe here, but it has several differential "blind spots" that can cause it to miss crossings. It is not easy to tell *a priori* how often these crossing detection gaps would cause them to miss a crossing, so we have done a couple runs with our codes in which we tried to mimic the "blind spots" of their crossing detection scheme. In comparison with runs of our code with the same initial conditions we have found that their scheme missed 62% of all the crossings, but only 13% of the crossings between different strings. These "blind spots" caused the mean loop size to be a factor-of-4 larger than the mean loop size in the control runs without the "blind spots," but the corresponding error in the evolution of the long-string density was too small to explain the factor-of-4 difference in our values for the long-string density

D. Tests of the evolution of isolated loops

1. *Tests in flat space*

One of the simplest tests one can apply to test the evolution algorithms described in the previous sections involve the evolution of a single loop in flat spacetime where the real trajectory is periodic. This test is not very interesting for code II because it is exact in flat space, but it is an important test of code I. We will follow the evolution of a loop over many periods to see how errors propagate. It is convenient to select a loop from Turok's two-parameter family of loop solutions[44] because these loops have a simple analytic solution. We have plotted in Fig. 3 the X-Y coordinates of one of these loops [with parameters $\alpha=0.4$, $\cos(\phi)=0.5$] every five and a half periods over a span of more than 100 periods. (The period of a loop is $L/2$ where L is the length of a loop.)

The left plot corresponds to the computed evolution when 128 sampling points are used (case 1), and the right one when 32 sampling points are used (case 2); the solid line outlines the initial configuration which is also the solution given by code II for integral periods. In both cases, the time step Δt was 0.8 times the maximum value allowed by the Courant condition ($c\Delta t_{\max}$=distance between neighboring points $=\epsilon d\sigma$). Even though errors do build up in the latter case (which has a larger time step), the loop nevertheless retains its general shape (and thus its cross section for reconnection). A measure of the error is given by $\delta\equiv\langle(\mathbf{r}-\mathbf{r}_a)^2/\mathbf{r}_a^2\rangle^{1/2}$ where the subscript a denotes the value computed analytically. This error in-

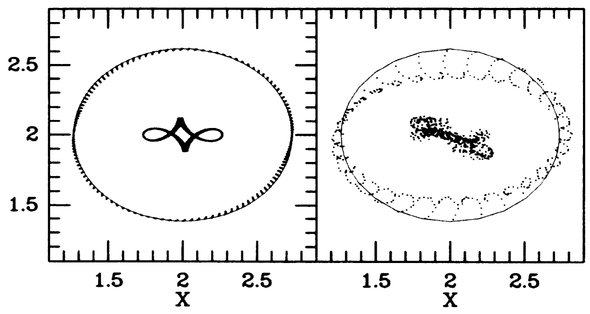

FIG. 3. The flat-space evolution with code I of loops from Turok's family of solutions with $\alpha=0.4$ and $\cos(\phi)=0.5$ using 128 sampling points (left) and 32 sampling points (right). The X-Y positions are plotted every 4.5 periods for more than 100 periods.

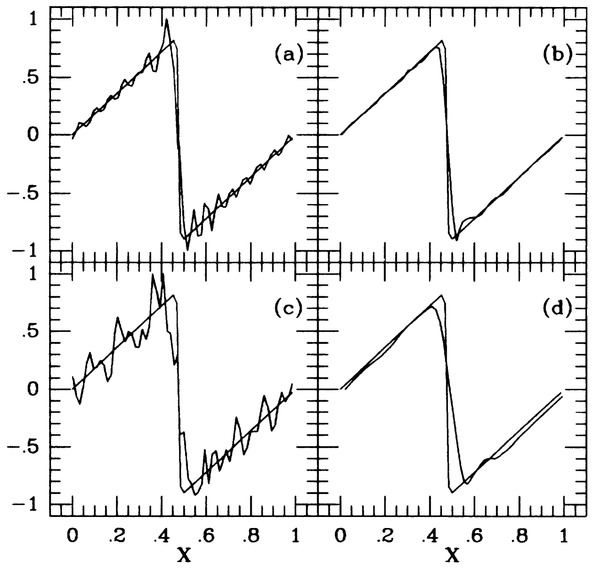

FIG. 4. The velocity component v_z vs x is plotted for a simple loop with a large amplitude kink for flat-space runs with and without our diffusion scheme. The evolution of v_z vs x after two periods is compared with (a) the initial profile without any diffusion and (b) with our scheme to invoke the diffusion only when it is needed. Plots (c) and (d) show the same thing after ten periods.

creases linearly with time and reaches $\delta_1 \simeq 0.6\%$ in the first case after 8000 steps, and $\delta_2 \simeq 5\%$ in the second case after 2000 steps. (This was 100 periods in both cases.) In both cases, the center-of-mass position and velocity were conserved to machine accuracy (i.e., $\Delta r_{\text{c.m.s.}}, \Delta v_{\text{c.m.s.}} \sim 10^{-6}-10^{-7}$ on a 32-bit Vax on which these tests were conducted for convenience; but our real experiments were performed on a 64-bit Cray-2). Even in the rather extreme case when only eight sampling points are used, we measure after 100 periods (500 steps only) $\delta_3 \simeq 40\%$. All these numbers are fairly representative of what we get in many different cases obtained by varying α and $\cos(\phi)$, and also by including a period two right mover. We should note that even though these performances might not seem very impressive for flat-space evolution (since it is trivial to build an exact evolution code for $c\Delta t \equiv \Delta\sigma$), the numbers we quote are for the realistic case where we used our actual mover with all its bells and whistles, and realistic values of every parameters.

As was already discussed in the subsection concerning the integration of the equations of motion, we introduce some numerical diffusion *when and where* the gauge condition (2.10) is violated by more than 5%. The effect of this prescription may be most easily visualized by considering an initially straight string in the x direction with a transverse velocity discontinuity along the z axis, which is the instantaneous superposition of a left-moving and a right-moving kink each with half the amplitude of the total discontinuity. Note that there has been some initial rounding of the kink to correspond with the amount of rounding that occurs in our intercommutation routine. As the string evolves, the two kinks separate. But because of the periodic boundary conditions, they superpose again when they have traveled a distance corresponding to half the box size. After an integer number of such periods, one can then compare the initial signal and the "filtered" one.

In Fig. 4 we have plotted the initial discontinuity and the transformed one after two periods (top illustrations), and after ten periods (bottom illustrations), when no smoothing is applied (left illustrations), and when we use our standard procedure (right illustrations). This test has used 64 sampling points, and the velocity difference on each side of the discontinuity was $1.8c$. Other tests on a wide variety of "kinky" loops yielded similar results. For comparison, if we were to evolve these strings with code II, we would find that the loop profile is exactly conserved since code II is exact in flat space. In fact, the discontinuity would be even sharper than the initial discontinuities in Fig. 4 because the intercommutation routine in code II does not smooth the kinks at all.

2. *Expanding universe tests*

All the single loop tests that have been described above have also been performed on code II, but the results were rather trivial because the evolution is always exact in flat space for any time step. This should not be taken as a weakness of the flat-space tests, but as a manifestation of the power of code II. Of course, it is also important to test the codes in an expanding universe as well. The difficulty here is that there are no known analytical solutions for string evolution in an expanding universe. Therefore we can only do tests between different numerical solutions.

One such test is shown in Fig. 5. The solid curve in this figure is the final position of a kinkless loop of length $l=0.5H_0$ evolved for an expansion factor of 4.0 using code I with 800 sampling points. The open circles give the position of the same loop evolved with only 20 points using code I, and the closed circles are for the same loop

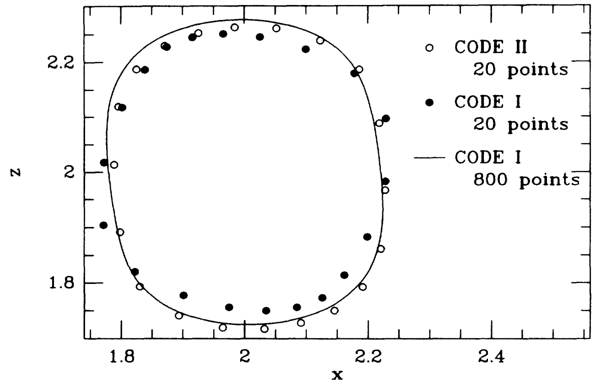

FIG. 5. The evolution of a kinkless loop in an expanding background is compared for three different runs. The solid line shows the X-Y positions for a run of code I with 800 points; the open circles show the configuration of the same loop evolved by code I with 20 points; and the filled circles are the same loop evolved by code II with 20 points.

evolved with 20 points using code II. For smooth loops, code I is essentially the same as the standard leapfrog which should be quite reliable particularly when we use 800 points. The fact that the 20 point code II run matches the 800 point code I run more closely than the 20 point run of code I indicates that code II is more accurate than code I even for loops that are very smooth.

It is also important to test the evolution of "kinky" loops in an expanding universe. We have performed numerous tests on loops with two kinks that are related to simple flat-space solutions. Figure 6 shows the evolution (using code II) of such a loop of length l from $\tau_0=0.38l$ to $\tau_0=2.5l$: an expansion factor of 6.6. The loop starts off at a high velocity and quickly expands. It then oscillates several times while shrinking in comoving coordinates (due to the expansion) and moving to the right. The majority of this evolution is just due to the changing superposition of the right- and left-moving waves, so $\mathbf{p}$ and $\mathbf{q}$ evolve very little throughout the run. Figure 7 shows the initial and final configurations of $\mathbf{p}(s_+)$. The evolution of $\mathbf{p}$ has been virtually monotonic, so $\mathbf{p}$ has really evolved very little during this test. The main effects have been the displacement of the points by about one point spacing along the original curve, and the reduction of the kink amplitude by a factor of about 0.86. $\mathbf{q}$ and ϵ also evolve in a similar manner. Thus code II has allowed us to replace the violent oscillations of $\mathbf{x}$ and $\mathbf{v}$ with the very slow and smooth evolution of $\mathbf{p}$, $\mathbf{q}$, and ϵ.

3. *Crossing detection tests*

We now turn to the question of the reliability and accuracy of our crossing detection and reconnection procedure. The reliability of the detection is not really an issue since our procedure is essentially exact. As a matter of fact, during the early testing period of these routines, we could not recover the phase-space diagram delimiting the region in $[\alpha,\phi]$ where intersections occur as initially determined by Turok.[44] We checked the discrepant cases by numerically solving the intersection equation $[\mathbf{x}(\sigma_1,\eta)=\mathbf{x}(\sigma_2,\eta)]$ which can be reduced to an eighth-order polynomial equation, and confirmed our findings. Our results were also independently confirmed by Chen, DiCarlo, and Hotes.[45] Moreover, we used this procedure to predict the occurrence as well as the detailed time and location of a crossing and the subsequent ones in a larger family obtained by adding a period-two right mover. Our routines never failed to detect an intersection, nor yielded spurious ones. Still, the position and time of the intersection are not infinitely precise, nor is our reconnection

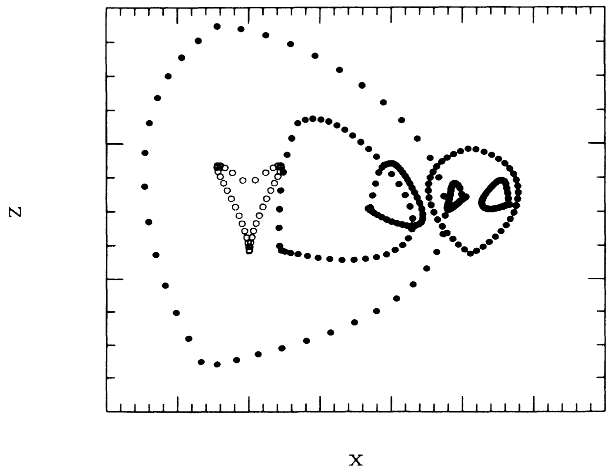

FIG. 6. The X-Y positions for a 40 point loop of length $2.63\tau_0$ at various times as the Universe expands by a factor of 6.6. This loop has two kinks.

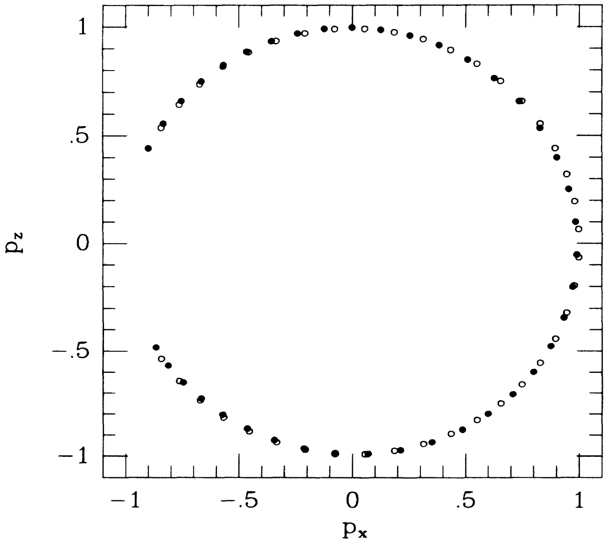

FIG. 7. The open circles show the initial values of p_x vs p_y for the loop plotted in Fig. 6, and the solid circles show the final values of p_x vs p_y.

procedure. When one loop fragments many times, a tiny perturbation might result in a fairly different final outcome, but this just means that the final pattern is quite sensitive to small changes in the initial loop trajectory. The comprehensive numerical tests that we have performed on our code (presented below) have demonstrated that this sensitivity to initial conditions does not effect the general statistical properties of the strings that are our primary interest.

E. Numerical characteristics of our production runs

We measure the physical size of our runs by the ratio of the initial horizon size to the persistence length ξ_0 of the strings in their Vachaspati-Vilenkin initial configuration. Since our simulation box has a fixed comoving size, we will refer to the box size in units of the comoving initial persistence length ξ_0^c. (ξ_0 is defined to be constant in physical units so $\xi_0^c = a\xi_0$ is constant in comoving units.) We will measure distances (or times) in terms of the horizon size $H = a\tau$ when we use physical units and in terms of τ (the conformal time) when we use comoving units. The physical density of the string network is adjusted by varying the ratio of the initial horizon size to ξ_0^c. We have found that we can run for a time interval equal to the box length (in units where $c = 1$) without encountering any difficulties due to the finite box size (see below), so most of our runs were run for this time interval.

Our largest simulation (done with code I) was in a box $36\xi_0^c$ on a side, with an initial horizon size $H_0 = 2ct_0 = 14\xi_0$ or $\tau_0 = 14\xi_0^c$. It was evolved for an expansion factor of 3.2 (more than a factor of 10 in physical time), so that the final horizon size was $\tau = 45\xi_0^c$. The total number of strings increased from about 1000 at the beginning to about 16 500 at the end. The strings were sampled by about 350 000 points (which corresponds to ten sampling points per initial correlation length ξ_0). Our highest resolution code I run was run on a $(21\xi_0)^3$ box and used 33 sampling points per ξ_0 for a total of 220 000 points. It was run from $\tau_0 = 8.75\xi_0^c$ to $\tau = 29.8\xi_0^c$: an expansion factor of 3.4. During the course of the run the number of strings grew from 200 to about 10 000. Each of these runs took about 40 CPU hours on a Cray-2.

Our largest matter-era run was done on a $(26\xi_0^c)^3$ box with ten sampling points per correlation length which corresponds to 130 000 points in all. It was run from $\tau_0 = 6.5\xi_0^c$ to $\tau = 28.6\xi_0^c$ which is an expansion factor of 19.4. The expansion factor accessible to our simulations is much larger in the matter era because $a \propto \tau^2$ in the matter era as opposed to $a \propto \tau$ in the radiation era. The main drawback of this is that $\epsilon \propto a^{-1}$ according to Eq. (2.3), so that the time steps in the matter era become much smaller than in a radiation-era run that has run for a similar conformal time (τ) interval. Furthermore, the matter-era string network at scaling does not chop itself into small loops nearly as fast as a radiation-era network does. This tends to slow down the code because the most inefficient part of the crossing detection is the detection of crossings involving the long strings. For these reasons, this run took almost 50 Cray-2 CPU hours.

This situation has been substantially improved with code II. We did a similar run on a $26\xi_0^c$ box with six sampling points per ξ_0 on each of the three grids ($s_\pm$ and σ). As we discussed above, this is considerably better resolution than the one we get with ten points per ξ_0 with code I. This run went from $\tau_0 = 6.5\xi_0^c$ to $\tau = 26\xi_0^c$ for an expansion factor of 16, and it took only 10 Cray-2 CPU hours. Allowing for the somewhat smaller expansion factor for this run compared to the code I run as well as hardware improvements which occurred between the runs, the code II run is a factor of 3 or 4 faster. The main reason for this reduced run time is that the time step for code II tends to be larger (i.e., closer to the Courant stability limit). This is due to a more flexible method for changing the time steps, and the larger separation between sampling points.

For the radiation-era runs there is not any spectacular gain in time used by code II, but there are some significant differences between the radiation-era runs with the two codes. The largest radiation-era code II runs that we have performed to date used a $(28\xi_0^c)^3$ box with ten points per ξ_0 on each grid for a total of $\sim 160 000$ points initially on each grid. These runs took about 25 CPU hours each which is about the same as a code I run with 15 points per ξ_0. The largest run went from $\tau_0 = 6.5\xi_0^c$ until $\tau = 42.0\xi_0^c$ or an expansion factor of 6.5 (a factor of 42.45 in real time). During this run the number of loops grew from about 500 to over 85 000. 75 000 of these loops are ones that have been produced by the flat-space fragmentation routine which are too small to include in the code for further evolution. For comparison the number of loops produced in the code I run with 15 points per ξ_0 is about 20 000.

III. EVOLUTION OF THE LONG-STRING NETWORK: NUMERICAL RESULTS

A. The scaling solution in the radiation era

The main result of our first paper[33] was strong evidence for the existence of a scaling solution in the radiation era. The major difficulty that we encountered with our early simulations was the dependence of our results on the minimum size we allowed for small loops. This did not cast any doubt on the *existence* of a scaling solution, but it did make the determination of the parameters of the scaling solution much more difficult. In this section, we will see how this situation has been greatly improved with code II.

As discussed above, the basic objective of our simulations is to study the properties of the scaling solution. Our basic strategy is to start with some initial state which is realistic (i.e., Brownian) on very large scales and which is not too far from the scaling solution. We then follow the evolution in time as the simulated network relaxes to the scaling solution. The question of how long this relaxation takes depends on what scale we are interested in. Figure 8 shows snapshots of half-horizon-sized fractions of one of our simulation boxes at several times during the course of a simulation run. Horizon-sized boxes were chosen because if the evolution is described by a scaling solution, the string network should always look the same (statistically speaking) on the horizon scale. Since these pictures are of a fixed volume in horizon-sized units while our box size has a fixed comoving volume, each successive picture includes a larger fraction of the box size. It is also clear that our resolution (in horizon units) increases dramatically throughout the run. This is one reason why the distribution of very small loops can never reach scaling during the run. One might hope that scaling might be achieved for somewhat larger loops, but this

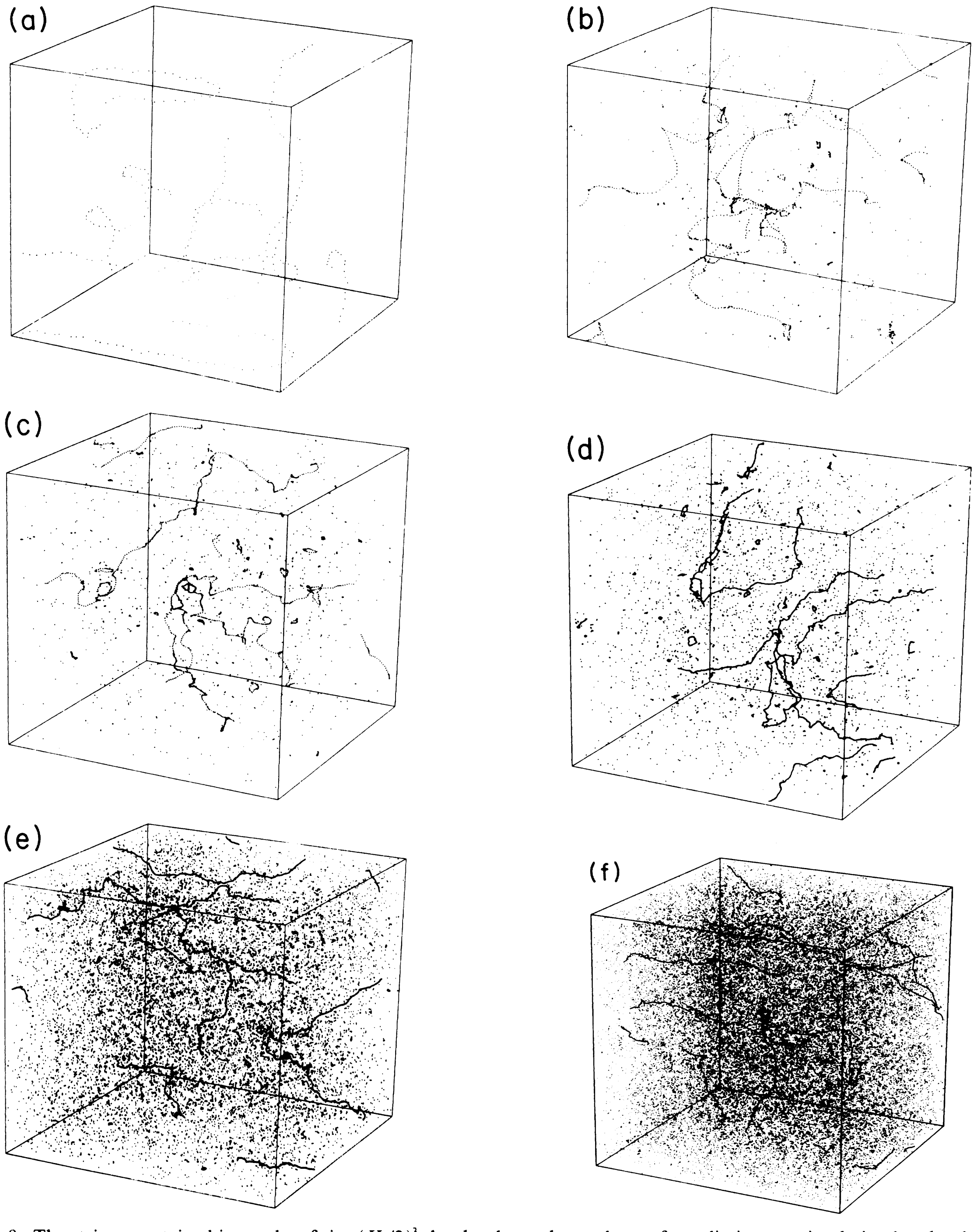

FIG. 8. The strings contained in a cube of size $(H/2)^3$ that has been chopped out of a radiation-era simulation box has been plotted at six different times. Each point represents a single point on the σ lattice. The expansion factor and fraction of the total box shown are (a) $a=1.01$, 0.16% of the box, (b) $a=1.81$, 0.92% of the box, (c) $a=2.30$, 1.91% of the box, (d) $a=3.12$, 4.75% of the box, (e) $a=4.93$, 18.6% of the box, (f) $a=6.47$, 42.3% of the box.

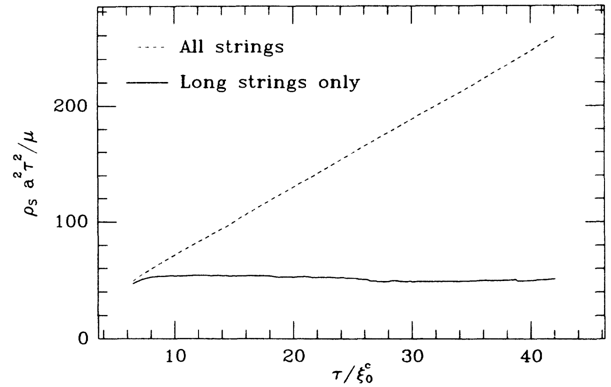

FIG. 9. Normalized energy density in strings $\rho_s H^2/\mu$ as a function of the conformal time τ in units of the initial comoving correlation length ξ_0^c. The upper curve represents all the strings in the simulation, and the lower curve is the density in strings with energy $E > H\mu$.

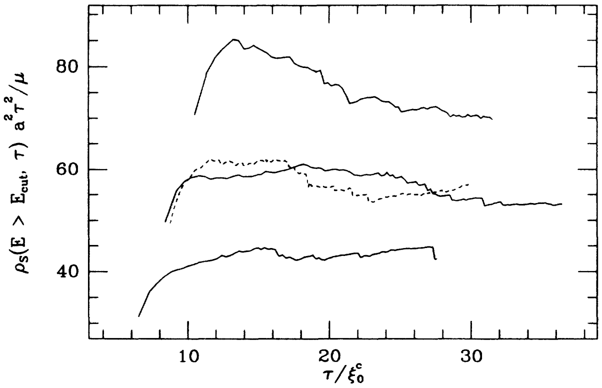

FIG. 10. The relaxation to scaling of the normalized long-string energy density $\rho_{LS}H^2/\mu$ for three runs of code II (solid lines) with different initial densities is plotted vs conformal time τ in units of the initial correlation length. (Long strings are defined as those with energy $E > H\mu$.) The dashed line is the highest-resolution run of code I.

is quite difficult to judge from Fig. 8 alone because of the preponderance of very small loops.

It is only for the infinite strings that Fig. 8 seems to show clear evidence of scaling. The total length of the long strings crossing a horizon volume settles down fairly quickly to the scaling value although the small-wavelength structure of the long strings relaxes much more slowly (more about this later). The scaling behavior of the long strings can be seen more quantitatively in Fig. 9. The solid curve in Fig. 9 represents the energy density in long strings ρ_{LS} multiplied by the factor $H^2/\mu \equiv a^2\tau^2/\mu$ plotted versus conformal time τ. This quantity $\rho_{LS}H^2/\mu$ is just the total proper length (length in the rest frame of the string) in units of H of all the long strings in a horizon volume H^3. Long strings are defined to be strings with a proper length longer than H. Since there are rarely any loops with a size close to H, the length in long strings is very insensitive to the precise numerical value used to distinguish between long strings and loops. In a scaling solution, the total number of long strings crossing a horizon volume and hence $\rho_{LS}H^2/\mu$ should be constants, so the flatness of the solid curve in Fig. 9 indicates scaling. Note that the x axis of Fig. 9 is in conformal time units so that the run shown in Fig. 9 ran for a factor of 4.3 in expansion or 18.5 in real time.

The dashed curve in Fig. 9 gives the total energy density of all the strings in our computational box multiplied by the same scale factor (H^2/μ). The difference between the dashed and solid curves is due to loop production by the long strings, so Fig. 9 provides graphic evidence that loop production is an efficient energy-loss mechanism for the long strings. In the real Universe, the dashed curve would also become flat in a scaling solution, but our simulation does not reflect this because we have not included the gravitational radiation energy-loss mechanism for the loops.

Figure 10 shows the relaxation of the long-string energy density to the scaling solution value for several runs of code II. The dashed curve is the highest-resolution run that was done with code I, so it is clear that the two codes give quite similar results. A better comparison of the difference between the results from our two codes can be seen from Fig. 11 which shows the evolution of ρ_{LS} for two pairs of runs with identical initial conditions and numerical parameters. The solid lines are the results obtained with code II, and the dashed lines are from code I. The curves sloping upward are $(28\xi_0^c)^3$ runs with a small loop cutoff of ξ_0, and the lower, more horizontal curves are from $(21\xi_0^c)^3$ runs with a cutoff of $\sim 0.14\xi_0$.

Thus, it is clear that the results from our different codes are very similar. This fact and the fact that code II has corrected some suspected systematic errors that we feared might have affected the results from code I have led us to substantially reduce the estimated errors on our value for ρ_{LS}. Our new value for the radiation-era scaling solution is $\rho_{LS} = (52 \pm 10)\mu/H^2$. [We should note that we now use H as our unit of length rather than t which was used in Ref. 33. In the old units our new value is $\rho_{LS} = (13 \pm 2.5)\mu/t^2$.] Although the error bars we have quoted for our new value are completely contained in the

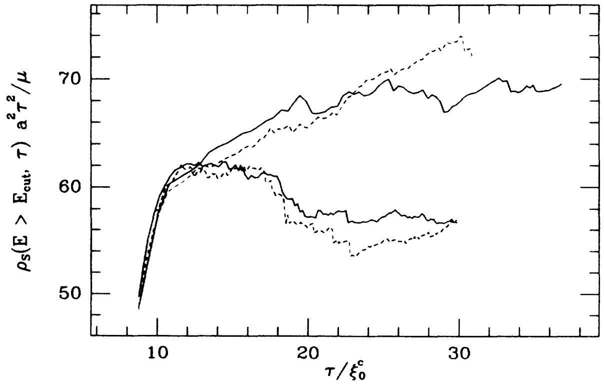

FIG. 11. A comparison of the evolution of $\rho_{LS}H^2/\mu$ for runs with different cutoffs done with the different codes. The solid lines are for code II and the dashed lines refer to runs of code I.

error bars given in Ref. 33, our central value for $\rho_{\rm LS}H^2/\mu$ is a bit lower than the value $\rho_{\rm LS}=(80\pm40)\mu/H^2$ reported in Ref. 33. This may seem a little puzzling since the results from our two different codes are much closer than this. The reason that we reported such a high central value for $\rho_{\rm LS}$ in Ref. 33 was that we suspected that a systematic error having to do with our velocity averaging scheme might have a significant effect on our results. The new results from code II have shown that this is not the case, so we have reduced both our estimated errors and the central value for our estimate of $\rho_{\rm LS}$.

B. Tests for numerical effects in the evolution of $\rho_{\rm LS}$

In Ref. 33, we indicated that we had studied the effects of various numerical parameters such as the cutoff on small loops, the box size, and the density of sampling points along the string. We found that only the small loop cutoff had a significant effect on our results. The small loop cutoff is enforced in our simulations by only allowing loops to be formed when they have more than a fixed number of points (ranging from 3 to 15). Because we also vary the density of sampling points of the string (the number of grid points per ξ_0), it is convenient to refer to the cutoff as a fixed fraction of $\xi_0=$(minimum No. of points per loop)/(No. of points per ξ_0). Our study of loop production from the long strings has indicated that the rate of loop production does not depend on the small loop cutoff, but that the rate at which energy is returned to the long strings via the reconnection of loops does depend on the cutoff. The reason for this is that, as was emphasized in Ref. 25, the number of small loops reconnecting to the long strings is roughly independent of the loop size for a given total string length in loops, but the amount of energy returned to the long-string network is proportional to the loop size. When a relatively large lower cutoff is used, a significant number of strings are artificially prevented from fragmenting into smaller loops. This increases the amount of energy returned to the long strings via reconnections, and therefore decreases the efficiency of loop production.

This is the reason why the numerical smoothing of the AT code has such a drastic effect on their value for $\rho_{\rm LS}H^2/\mu$ at scaling. As is shown in Appendix A, the structure on scales smaller than the scale length of the long strings in their simulations is smoothed out quite efficiently. Thus, the loops that break off the long strings in their simulations tend to be much too smooth and break up into fewer stable loops[46] than they should. (The "blind spots" in their crossing detection scheme could also play a role in this.) This means that their loop distribution has far too many large loops so that the reconnection of loops back onto the long-string network can transfer a significant amount of energy back to the long-string network. In fact, they find that about 50% of the energy lost to loops in their simulations is returned to the long strings via reconnections. In our high resolution, small cutoff simulations, this figure is about 10% while in some of our large cutoff run it is somewhat higher than 30%. Thus, AT's results are what we would expect from our code if we had run with an even larger lower cutoff than any of the runs that we have done.

We also believe that this sort of cutoff effect is responsible for the relatively small difference between our results and those of Allen and Shellard (AS). They have done the majority of their runs with a cutoff of about $0.5\xi_0$ and obtained $\rho_{\rm LS}H^2/\mu=64\pm16$. This is noticeably higher but still in agreement with our result of $\rho_{\rm LS}H^2/\mu=52\pm10$. The runs we have done with a cutoff of this size seem to approach scaling near the upper limit of our error bars, which suggests that the difference between our central values may just be due to a small residual cutoff effect in their code.

With the improved resolution of code II, we have been able to set the cutoff small enough so that this cutoff effect has disappeared. Furthermore, our flat-space fragmentation scheme has allowed us to do runs with an effective cutoff that is very much smaller than the values which can influence the long-string evolution. The cutoff must be set to be small enough so that the amount of energy returned to the long strings via loop reconnections has become independent of the lower cutoff. This can be seen in Fig. 12. The top curve (with alternating long and short dashes) of this figure is a run with a cutoff of ξ_0. It is apparently heading toward a cutoff-dependent scaling solution at $\rho_{\rm LS}\sim100\mu/H^2$ like the runs of code I with the same cutoff that were shown in Fig. 2 of Ref. 33. The solid curve in Fig. 12 has a cutoff of $0.3\xi_0$. The curves made of long and short dashes correspond to runs in which the flat-space fragmentation scheme has been turned on. This means that their effective lower cutoff is smaller than the actual cutoff because small loops can fragment into arbitrarily tiny loops (although there is limit on the number of tiny loops it can fragment into based on the number of grid points on the initial loop's s_+ and s_- lattices.) The cutoffs for these runs are $0.33\xi_0$ (the short-dashed curve) and $0.13\xi_0$ (the long-dashed curve), but their effective cutoffs are considerably smaller. In any event, the cutoffs for two of the runs at the bottom

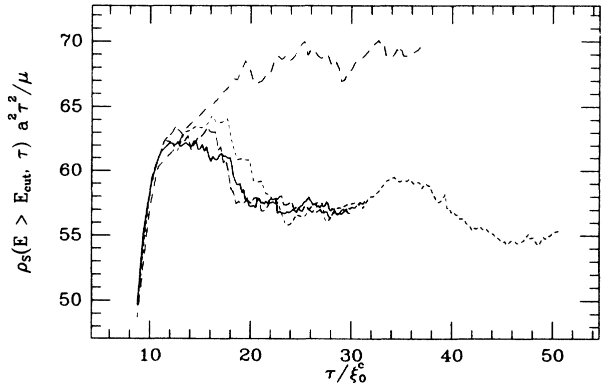

FIG. 12. The effect of the small loop cutoff on the evolution (using code II) of long strings is examined by comparing the evolution of $\rho_{\rm LS}H^2/\mu$ for runs with sharp cutoffs at ξ_0 (the upper curve with alternating dots and dashes) and $0.3\xi_0$ (the long-dashed curve) with runs with cutoffs of $0.33\xi_0$ (the short-dashed curve) and $0.13\xi_0$ (the solid curve) for the large loops only.

of Fig. 12 differ by a factor of >2.5. This makes it quite clear that the cutoff effect essentially disappears for cutoffs of $0.3\xi_0$ or smaller.

Figure 12 is also useful for studying the effect of the density of sampling points on our results. The two dashed curves at the bottom of Fig. 12 differ in their sampling density by a factor of 2.5. The long-dashed curve is for a run with 15 points per ξ_0 for each of the three lattices ($s_\pm$ and σ), and the short-dashed curve is for a run with six points per ξ_0. Thus, the evolution of $\rho_{\rm LS}$ is unaffected by sampling densities as small as six points per ξ_0 (with code II).

The effect of varying the sampling density for code I is shown in Fig. 13. The top two curves in this figure were generated using a small loop cutoff equal to ξ_0. The top dashed curve used five points per ξ_0, and the top solid curve used ten points per ξ_0. The lower curves used a cutoff of $0.3\xi_0$ and sampling point densities of 20 (for the lower solid curve) and 10 (the lower dashed curve) sampling points per ξ_0. When the small loop cutoff is large, a sampling density as small as five points per ξ_0 seems to work fine, but when the cutoff is small, a small systematic error seems to develop with ten points per ξ_0. This systematic error is probably caused by the numerical diffusion that we have introduced into code I in order to prevent the development of instabilities at the kinks. With a small cutoff, the smoothing of the strings on small scales will inhibit the production of very small loops which will decrease the efficiency of loop production as an energy-loss mechanism for the long strings. Since this smoothing is confined to a few lattice lengths or so, it becomes less of a problem when a larger sampling density is used.

Another numerical parameter which could strongly influence the evolution of $\rho_{\rm LS}$ is the finite box size. Since we are using periodic boundary conditions, if we attempt to run a simulation for too long a time then the horizon H will grow to be much larger than the size of the box. When this happens, the string network in our periodic box no longer resembles a string network in the real Universe. One possibility is that some of the long strings can become tightly stretched across the box with nonzero winding number. This would always occur if our initial state had an overall winding number, but the Vachaspati-Vilenkin procedure that we employ always generates states of zero winding number. Nonetheless, it is possible to have two or more strings each with net winding number (but with zero total winding number) which become frozen in place and start stretching with the expansion so that $\rho_{\rm LS} \sim a^{-2}$ as $t \to \infty$. The other possibility is that all the long strings would annihilate leaving $\rho_{\rm LS}=0$ forever.

Because signals on the string cannot travel faster than the speed of light, a string network cannot "know" that it is confined to a finite box until the simulation has been run long enough so that the light travel distance $(\tau-\tau_0)$ equals the box size. In fact, one might hope that we could run considerably longer than $\tau-\tau_0$ since the scale length of the long strings is always considerably smaller than τ.

Figure 14 shows the evolution of $\rho_{\rm LS}$ for several runs of code I with box sizes of $(16\xi_0^c)^3$, $(21\xi_0^c)^3$, and $(36\xi_0^c)^3$. The initial horizon size (or conformal time) τ_0 in each of these runs was $12.6\xi_0^c$. Thus, according to the argument given above, the $(16\xi_0^c)^3$ runs should be free of finite box effects until at least $\tau=28.6\xi_0^c$, and the $(21\xi_0^c)^3$ run should not have any problems until at least $\tau=33.6\xi_0^c$. It is clear from Fig. 14 that finite box-size effects are not important when $\tau-\tau_0<$ the box size. In fact the $(16\xi_0^c)^3$ runs seem to show only a small deviation from scaling at $\tau-\tau_0=27.4\xi_0^c$ which is 1.7 times the box size. It is not clear whether this small deviation is due to boundary condition effects or is just a manifestation of the larger statistical fluctuations to be expected when $\tau \gg$ the box size. Figure 15 shows two $(21\xi_0^c)^3$ runs which were run for an interval of $\tau-\tau_0$ equal to twice the box size. The solid curve shows no apparent effect of the finite box, but

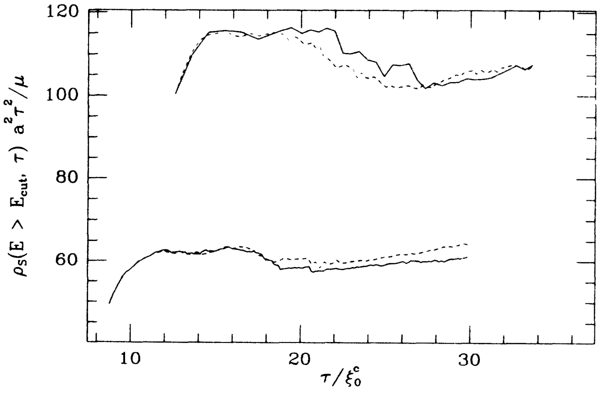

FIG. 13. A comparison of the evolution of $\rho_{\rm LS}H^2/\mu$ for pairs of different runs of code I which differ only in the number of sampling points per initial correlation length ξ_0. The top two runs have a small loop cutoff of $l_c=\xi_0$ and are sampled with five and ten points per ξ_0. The bottom two runs have $l_c=0.3\xi_0$ and are sampled with 10 and 20 points per ξ_0. The higher-resolution runs are shown by the solid curves.

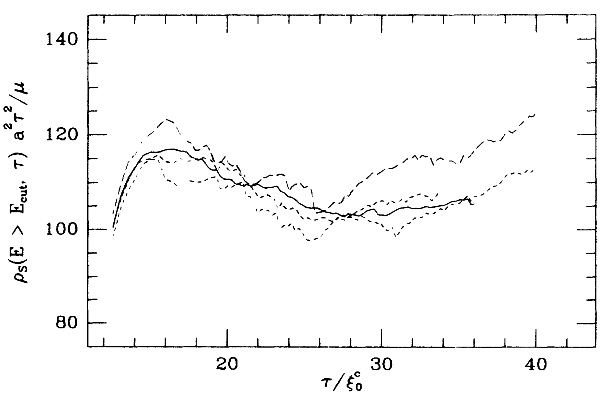

FIG. 14. The effect of the periodic boundary conditions and the finite box size on the evolution of $\rho_{\rm LS}H^2/\mu$ for runs of code I. The solid curve is from a run on a $(36\xi_0^c)^3$ box, the short-dashed curve comes from a run on a $(21\xi_0^c)^3$, and the long-dashed lines come from $(16\xi_0^c)^3$ runs.

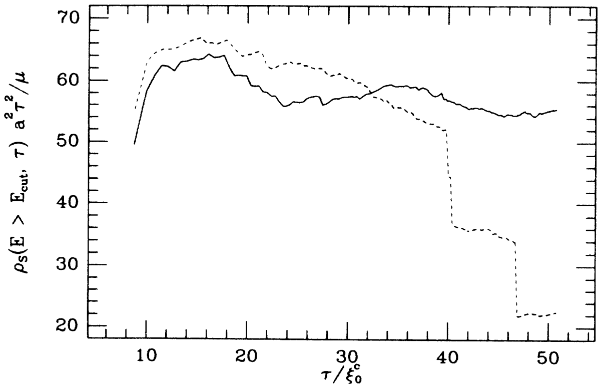

FIG. 15. The evolution of $\rho_{LS}H^2/\mu$ for two long radiation-era runs using code II. These runs were done on $(21\xi_0^c)^3$ boxes, they were run for twice the usual time interval: $\tau-\tau_0=42\xi_0$, an expansion factor of 5.8. The initial conditions for these runs were only slightly different, but the run with the dashed curve seems to have experienced spurious energy loss due to finite box-size effects.

the run corresponding to the dashed curve seems to have lost most of its long-string energy through spurious fragmentation of the long strings when the box length was well inside the horizon. Our conclusion from these tests is that we are certainly safe in running the simulation for a time interval equal to the box size and probably a bit longer. The exact point where finite box-size effects start to dominate is not clear due to the poor statistics in boxes with $\tau \gg$the box size, but strong finite box effects do occur (in some cases) when $\tau-\tau_0$ is twice the box size.

C. The effect of small-scale structure on the evolution of ρ_{LS}

Although the details of the development of small scale structure on the long-string network will be dealt with in great detail in the sequel to this paper,[30] we must demonstrate that our long-string evolution results are not sensitive to the details of the small-scale structure in order to be confident that the long-string results are correct. If we start with smooth initial conditions, the natural tendency is for the amount of small-scale structure to build up as more and more kinks are produced over the course of the run. There are a couple physical mechanisms that might serve to slow or stop the buildup of small-scale structure such as stretching by the expansion and the tendency for very "wiggly" pieces of string to preferentially chop off the network as loops. (Gravitational-radiation back reaction is a third mechanism which could smooth the small-scale structure, but it does not operate on the scales available in the simulation.) As we have shown in Appendix B, however, the expansion of the Universe reduces a kink amplitude as $\sim a^{-0.14}$, so this is not very effective at smoothing out the strings. The effectiveness of the second smoothing mechanism is rather difficult to judge except by observing what happens in the simulations. We find that, when the strings are studied in units with a constant physical size, the strings are rapidly smoothing themselves on all scales (after an initial transient period when the small-scale structure first appears). However, when we change to units that grow with the horizon, we find that the small-scale structure extends to smaller and smaller scales as the simulation runs and our resolution improves with respect to the horizon. (The fraction of the total string energy in the small-scale structure remains constant, however.) Since this small-scale structure never reaches scaling, it is important to show that it does not affect the large-scale structure of the strings.

One simple way to check that the small-scale structure does not influence the evolution of ρ_{LS} is to compare the short-dashed and long-dashed curves in Fig. 12. Recall that these runs started with identical initial conditions but that the short-dashed curve run had only six points per ξ_0 per lattice while the long-dashed curve run had 15. Toward the end of the run, enough kinks have built up on the runs with six points per ξ_0 so that many pairs of the neighboring kinks must be merged to form single kinks. This happens much less frequently with the 15 points per ξ_0 run, but we can apparently see no effect of this in the evolution of ρ_{LS}.

A much more drastic variation of the small-scale structure on the long strings can be obtained by varying the initial conditions. As we have mentioned in Sec. II A, we have added an option to our initial conditions that allows us to start with random velocities at each point on the strings. We select these velocities completely at random with a fixed magnitude which we can vary between runs. In order to satisfy the gauge condition (2.1), the component of velocity parallel to x' must be discarded. This reduces the average random velocity by $\sqrt{2/3}$. The run displayed in Fig. 8 started with a random velocity amplitude of 0.9 (before applying the gauge condition constraint), and Fig. 8(a) shows the configuration after one time step when this short-wavelength structure has just begun to be transferred to the string positions. Figure 16

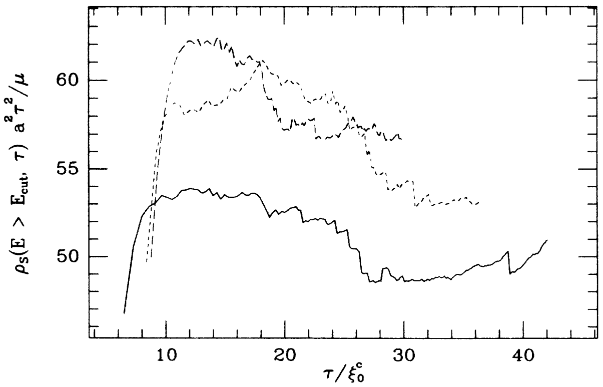

FIG. 16. The evolution of $\rho_{LS}H^2/\mu$ for runs of code II with varying amplitudes of initial short-wavelength structure. The initial random velocities at every point have an average amplitude of 0.73 (for the solid curve), 0.41 (for the short-dashed curve), and 0 (the long dashed curve).

shows the evolution of $\rho_{LS}H^2/\mu$ for three runs with varying amplitudes of this initial short-wavelength structure. Although there are some systematic differences early in these runs due to the differing initial conditions, all the runs seem to be approaching quite similar asymptotic values of $\rho_{LS}H^2/\mu$. The differences between these runs are no larger than the differences we have observed between runs with the same amount of initial small-scale structure but slightly different initial conditions. We therefore conclude that our long-string results *are not*

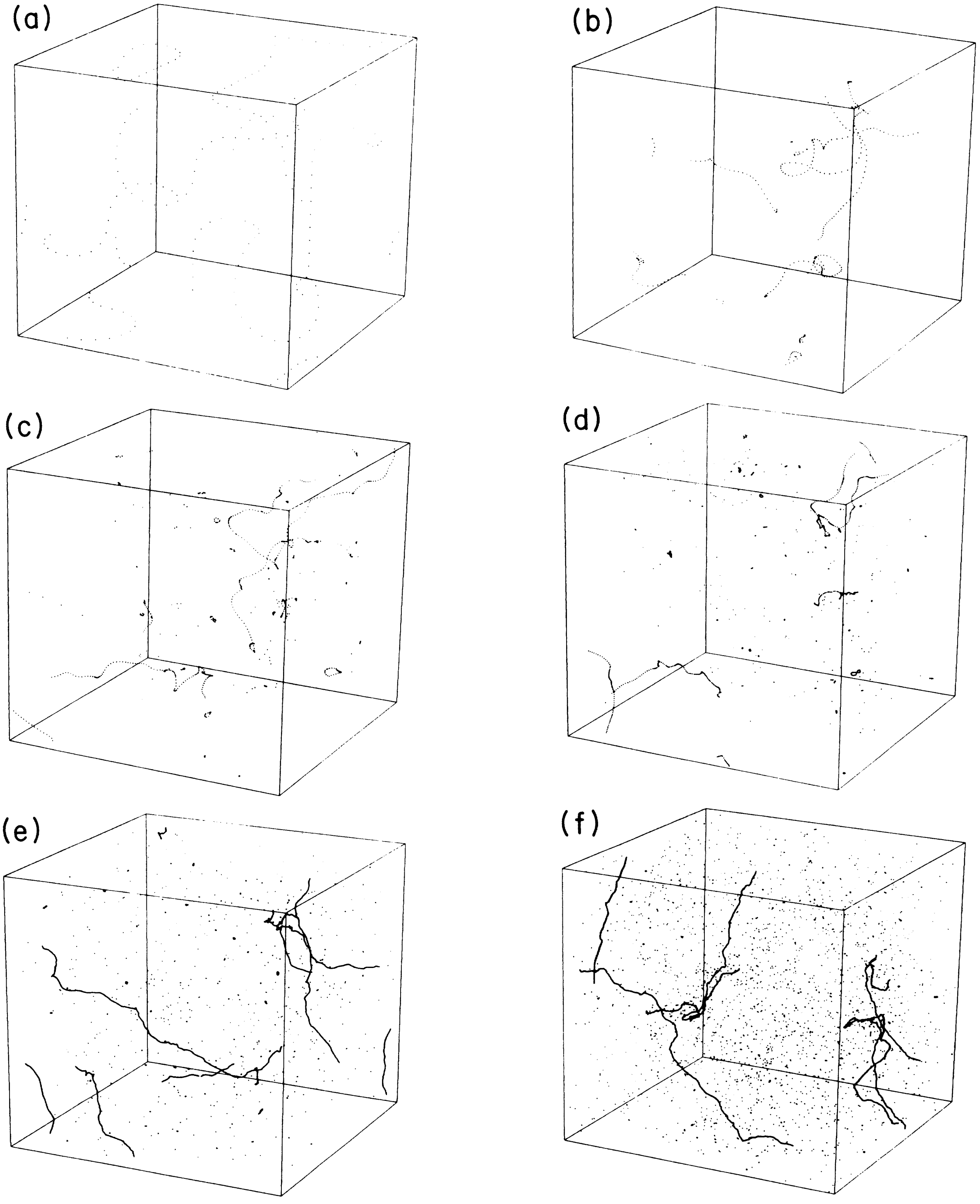

FIG. 17. The strings contained in a cube of size $(H/2)^3$ that has been chopped out of a matter-era simulation box has been plotted at six different times. Each point represents a single point on the σ lattice. The expansion factor and fraction of the total box shown are (a) $a=1.00$, 0.20% of the box, (b) $a=1.98$, 0.54% of the box, (c) $a=2.95$, 0.99% of the box, (d) $a=4.32$, 1.75% of the box, (e) $a=7.65$, 4.13% of the box, (f) $a=16.0$, 12.5% of the box.

influenced by the small-scale structure on the long strings as long as we run with sufficient resolution ($\geq$ six points per ξ_0 per lattice for code II) and a sufficiently small lower cutoff on loop size ($\leq 0.33\xi_0$). These conclusions are further strengthened by our discussion of loop-production mechanisms in Ref. 30.

D. Evolution in the matter and transition eras

1. The matter era

Figure 17 shows a series of boxes of size $H/2$ that have been taken out of one of our matter-era simulations that ran for an expansion factor of 16. There are a few obvious differences between Fig. 17 and the radiation-era string network shown in Fig. 8. The matter-era string network has much less small-scale structure, and it produces many fewer loops. The lower loop-production amplitude can also be seen in Fig. 18 which is the matter-era analog to Fig. 9. The reason for the smaller amount of loop production in the matter era is that the energy density of a noninteracting network of long strings scales as $a^{-3+\delta}$ where $0<\delta \ll 1$. In the radiation era, loop production must be sufficient to make $\rho_{LS} \propto a^{-4}$ while in the matter era we only require $\rho_{LS} \propto a^{-3}$ to have a scaling solution. Clearly, a scaling solution in the matter era requires much less loop production. Less loop production also implies fewer total interactions between the strings, so it is not surprising that Fig. 17 shows much less small-scale structure than Fig. 8.

Figure 19 shows the relaxation of ρ_{LS} toward the scaling solution value versus conformal time τ/ξ_0^c for runs with different initial long-string densities. We have determined the scaling solution value to be $\rho_{LS}=(31\pm7)H^2/\mu$. The dashed curve shows a run of code I which seems to show a slight systematic difference toward the end of the run. We believe that this is probably just a fluctuation due to the statistically different initial conditions, but it could also be caused by systematic errors caused by the numerical diffusion scheme used in code I.

The effect of varying the small loop cutoff is shown in Fig. 20. Varying the cutoff has a considerably smaller effect in the matter era than the radiation era. Since the Universe expands faster and there are many fewer loops produced in the matter era, loop reconnection has a smaller effect on the evolution of ρ_{LS} than in the radiation era.

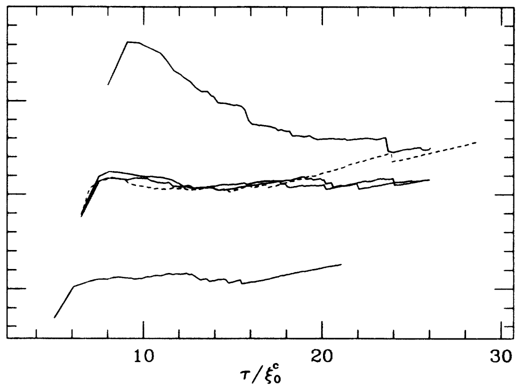

FIG. 19. The relaxation to scaling of the long-string energy density $\rho_{LS}H^2/\mu$ for 4 matter-era runs of code II and one matter-era run of code I (dashed curve) with different initial densities.

Figure 21 shows the effect on the evolution of ρ_{LS} of varying the amount of initial small-scale structure on the long strings. As in the radiation era, there is a noticeable difference in the transient behavior of the runs with different amounts of initial small-scale power. However, the difference always remains small, and the different runs seem to be approaching very similar values of ρ_{LS}. We therefore conclude that neither the small loop cutoff nor the amount of initial small-scale power on the long strings has a significant influence on our determination of the scaling solution value of $\rho_{LS}H^2/\mu$.

2. The radiation-matter transition

Before we discuss the evolution of the string network during the transition between radiation and matter domi-

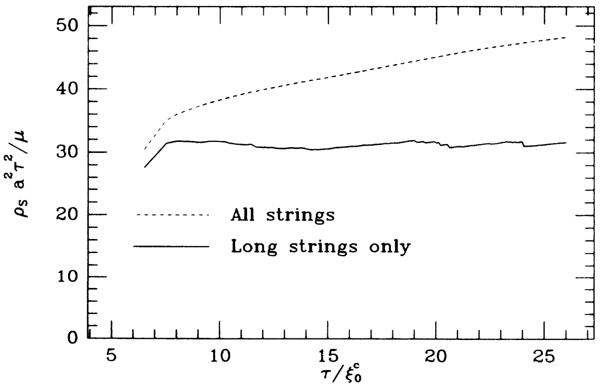

FIG. 18. Normalized energy density in strings $\rho_s H^2/\mu$ as a function of the conformal time τ in units of the initial comoving correlation length ξ_0^c. The upper curve represents all the strings in the simulation, and the lower curve is the density in strings with energy $E > H\mu$.

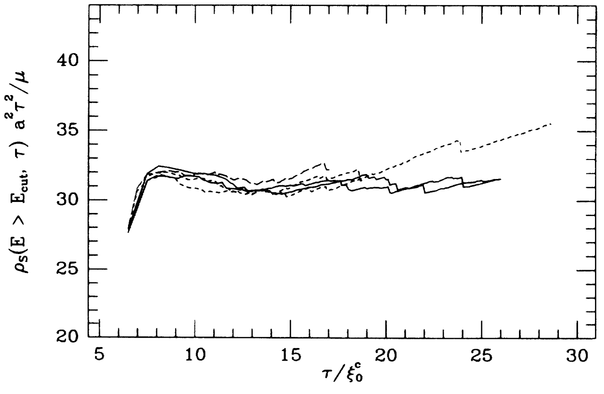

FIG. 20. The effect of the small loop cutoff on the evolution of ρ_{LS} in the matter era. The long-dashed curve shows the evolution of ρ_{LS} for a code I run with a cutoff of ξ_0, and the short dashes represent two code I runs with a cutoff of $0.5\xi_0$. The solid curves represent two code II runs with a cutoff of $0.33\xi_0$ which used the flat-space fragmentation routine.

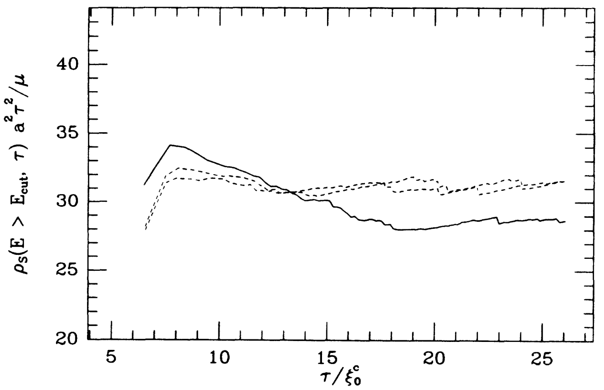

FIG. 21. The evolution of $\rho_{LS}H^2/\mu$ for matter-era runs of code II with differing amplitudes of initial short-wavelength structure. The initial random velocities at every point have an average amplitude of 0.41 (for the solid curves) and 0 (the dashed curve).

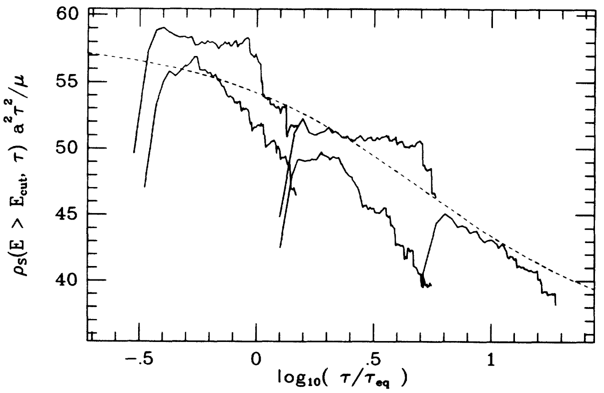

FIG. 22. The evolution of $\rho_{LS}H^2/\mu$ for six code II runs spanning the transition between the radiation- and matter-dominated eras is plotted vs the logarithm of conformal time. The dashed curve is a "Kibble model" fit.

nance, let us comment on the choice of a unit of length to measure the string density. We chose the horizon length H as our unit of length, so we measure ρ_{LS} in units of μ/H^2 in contrast with our first paper, where we measured ρ_{LS} in units of μ/t^2. AT (Ref. 31), on the other hand, have taken the Hubble radius $R_H = a/\dot{a}$ as their fundamental unit of length so they measure ρ_{LS} in units of μ/R_H^2. Although these three different units of length are all proportional to each other, the relative constants of proportionality are different in the radiation and matter eras. Thus, an unnatural choice of units could tend to exaggerate the changes in ρ_{LS} during the transition between radiation and matter dominance.

One might argue that the appropriate choice of units is the Hubble radius[31] because it is R_H^{-1} that actually appears in the damping term of the equation of motion (2.2). This would certainly be correct *if* the motion of the strings were dominated by the damping term, but it is not. The motion of the strings in both the radiation and matter eras seems to be closer to the free oscillation of strings in flat space than to motion that is dominated by damping. One indication of this is that the mean-square velocity of the string network in both the radiation ($\langle v^2\rangle = 0.43$) and matter ($\langle v^2\rangle = 0.37$) eras is much closer to the freely oscillating string limit ($\langle v^2\rangle = 0.5$) than to the heavy damping limit ($\langle v^2\rangle = 0$). Therefore we have the horizon size H as our length unit because it corresponds to the conformal time τ in comoving coordinates, and τ is the unit of time in Eq. (2.2). It is waves of wavelength H that have had time for a single oscillation.

Another indication that H is the proper choice of units is that the variation of ρ_{LS} is not very large when measured in these units. We find that the matter-era value of $\rho_{LS}H^2/\mu$ is 58% of the radiation-era value. If we had taken t as our unit of length this ratio would be 26% while with AT's choice of R_H as the unit of length it would be 14%.

The evolution of $\rho_{LS}H^2/\mu$ for six transition era runs is shown in Fig. 22. The earliest transition era runs shown here started at about $\tau = 0.3\tau_{eq}$ at a value of $\rho_{LS}H^2/\mu \sim 56-58$ which is somewhat larger than our central scaling solution value but well within the error bars. When the latest run ends at $\tau = 19\tau_{eq}$, $\rho_{LS}H^2/\mu$ has almost reached the range of acceptable matter-era values (31 ± 7), so these runs have covered most of the range of variation of $\rho_{LS}H^2/\mu$ during the transition. The midpoint of the transition is roughly $5\tau_{eq}$ which is also the midpoint of the transition for the scale factor power law:

$$a \propto \tau(1+\alpha\tau/\tau_{eq}) \ , \tag{3.1}$$

where $\alpha = (\sqrt{2}-1)/2 \approx \frac{1}{5}$, so there is little time to lag between the change of the expansion law and the response of the string network. Further discussion of string evolution in the transition era will follow in the next section after we have introduced the Kibble model.

IV. KIBBLE'S ONE-SCALE MODEL

A. The simplified Kibble model in the radiation and matter eras

In this section, we will compare our numerical results with the analytical model that has been developed by Kibble[18] to describe string evolution in an expanding universe. This model has been further developed by one of us[25,26] (D.P.B.), and variation of it has also been used by AT (Ref. 31) to try and fit their numerical results. We will see that Kibble's one scale model fits some of our results fairly well, but in other respects it seems to miss much of the important physics. This contradicts some of the claims of AT who found that Kibble's model seemed to fit their results very well.

The basic starting point for the Kibble model is the following equation for the energy density in long strings:

$$\dot{\rho}_{LS} = -2\frac{\dot{a}}{a}(1+\langle v^2\rangle)\rho_{LS} - \dot{\rho}_{\text{to loops}} + \dot{\rho}_{\text{from loops}} \ , \tag{4.1}$$

where $\langle v^2\rangle$ is a spatial average of v^2, and the dots refer to derivatives with respect to the conformal time τ. Equation (4.1) can be derived by integrating the ϵ evolution equation (2.3) over σ and adding the loop production and reconnection terms. The fundamental assumption made by Kibble is that there should be a single scale L which characterizes the long strings. This implies that

$$\rho_{\mathrm{LS}}=\frac{\mu}{L^2}\ , \tag{4.2}$$

and

$$\dot{\rho}_{\text{to loops}}=\frac{Ca}{L}\rho_{\mathrm{LS}}=\frac{Ca\mu}{L^3}\ . \tag{4.3}$$

[The factor of a appears in Eq. (4.3) because the time derivative of $\rho_{\text{to loops}}$ is with respect to conformal time.] Equation (4.2) has defined L so that a volume L^3 contains a total (proper) length L of string. Equation (4.3) can be regarded as the definition of the loop production parameter C, so the one-scale hypothesis of Kibble would imply that C is a constant independent of L. The scaling solution implies that L is proportional to $H=a\tau$. We should note that the value of C depends on our choice of time variable. If we had taken t or H as our time variable instead of τ, then the value of C in Eq. (4.3) would be different.

The chief difficulty with the analysis of Kibble's one-scale model has been the treatment of the $\dot{\rho}_{\text{from loops}}$ term. Bennett[25,26] has shown that the importance of this loop reconnection term depends quite sensitively on the shape of the loop-production function. In fact, as we have explained in the previous section, the difference between our results and those of AT can be explained largely by the difference in our loop-production functions and the different rates of loop reconnection onto the long strings that these loop-production functions imply. Nevertheless, *our* results indicate that loop reconnections do not play a very important role in long-string evolution. This conclusion is supported by the insensitivity of our long-string results to our lower cutoff (see Fig. 12) and to the "kinkyness" of our initial conditions (see Fig. 16). Both of these parameters have a large influence on the shape of our loop distribution but very little influence on the evolution of ρ_{LS}. This implies that our loop sizes are small enough so that reconnection is negligible.

Once we have dropped the $\dot{\rho}_{\text{from loops}}$ term from (4.1), it reduces to

$$\frac{\dot{L}}{L}=\frac{\dot{a}}{a}(1+\langle v^2\rangle)+\frac{Ca}{2L}\ . \tag{4.4}$$

If we set $L=\gamma H=\gamma a\tau$ so that γ is a constant at scaling,[47] then we obtain the scaling solution conditions

$$C_r=2(1-\langle v^2\rangle_r)\gamma_r \tag{4.5}$$

in the radiation era, and

$$C_m=2(1-2\langle v^2\rangle_m)\gamma_m \tag{4.6}$$

in the matter era. Note that if we set $\langle v^2\rangle$ to its flat-space value $\langle v^2\rangle=\frac{1}{2}$, then Eq. (4.5) implies $C_r=\gamma_r$ while Eq. (4.6) implies that $C_m=0$. This indicates a qualitative difference between the energy balance of the scaling solutions in the radiation- and matter-dominated eras. If we take the limit where the scale length of the long strings L is very much smaller than the horizon, then $\langle v^2\rangle$ approaches its flat-space value. In this limit, a finite amount of loop production would be necessary for scaling during the radiation era, but in the matter era, scaling can be achieved with no loop production at all. Realistically, we certainly do not expect loop production to cease in the matter era, so it must be that $\langle v^2\rangle$ is significantly less than $\frac{1}{2}$ during matter domination. This means that there will be a significant amount of string stretching to balance the energy loss to loop production during the matter era.

These simple arguments lead to the following qualitative predictions of the Kibble model[18,25,26]: (1) $\langle v^2\rangle$ will decrease during the transition from radiation to matter domination; (2) γ will increase during the transition because the scale length of the strings L must be close to the horizon size for string stretching to be important; (3) finally, one expects that C will probably decrease during the transition because loop production is not so important then.

Our radiation-era simulations indicate that $\gamma_r=0.139\pm0.013$ and $\langle v^2\rangle_r=0.43\pm0.02$ which implies that $C_r=0.16\pm0.02$. Similarly, our matter-era simulations give $\gamma_m=0.18\pm0.02$ and $\langle v^2\rangle_m=0.37\pm0.02$ implying $C_m=0.09\pm0.02$. Thus, the loop-production parameter C seems to vary by a factor of about 2 between the radiation and matter eras. This indicates that it is probably not a very good approximation to use the Kibble model to extrapolate through the transition between the matter and radiation eras.

In order to study the relaxation to the scaling solution from initial states with L larger or smaller than the scaling value, we can solve (4.4) if we take $\langle v^2\rangle$ to be a constant (which is a good approximation in the radiation era). The solutions are

$$\frac{\gamma-\gamma_r}{\gamma_0-\gamma_r}=\left[\frac{a}{a_0}\right]^{\langle v^2\rangle_r-1} \tag{4.7}$$

in the radiation era and

$$\frac{\gamma-\gamma_m}{\gamma_0-\gamma_m}=\left[\frac{a}{a_0}\right]^{\langle v^2\rangle_m-1/2} \tag{4.8}$$

in the matter era where $\gamma=L/H$ is no longer assumed to be a constant and a_0 and γ_0 are initial values. Figure 23 shows the evolution of γ vs $(a/a_0)^{\langle v^2\rangle-1}$ for several radiation-era runs. The simplified one-scale model predicts that these curves should be linear and intersect at $(a/a_0)^{\langle v^2\rangle-1}=0$ $(a\rightarrow\infty)$ at the scaling solution value $\gamma_r\simeq0.14$. Except for an initial transient due to the peculiarities of the initial conditions, these curves seem to fit straight lines intersecting at $a\rightarrow\infty$ very well, so this simplified version of Kibble's one-scale model seems to accurately describe the relaxation to scaling in the radiation era.

The situation is somewhat different in the matter era as can be seen from Fig. 24. The simplified one-scale model predicts that the approach to scaling should be much slower than in the radiation era: $\gamma-\gamma_m\sim a^{\langle v^2\rangle-1/2}\simeq a^{-0.13}$ for $\langle v^2\rangle\simeq0.37$ in the matter era versus $\gamma-\gamma_r\sim a^{\langle v^2\rangle-1}\simeq a^{-0.57}$ for $\langle v^2\rangle\simeq0.43$ in the radiation era. In Fig. 24 this translates into the prediction that the curves should be straight lines intersecting at $(a/a_0)^{\langle v^2\rangle-1/2}=0$. The curves in Fig. 24 seem to be con-

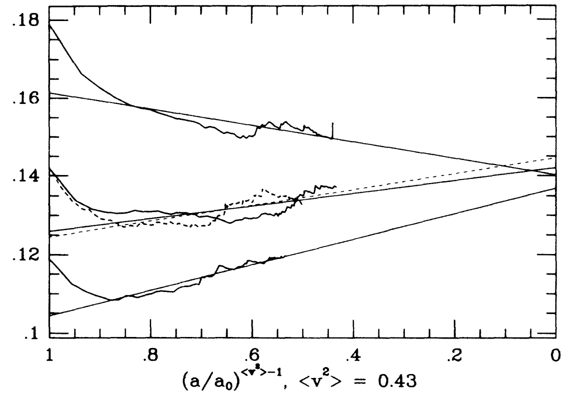

FIG. 23. The evolution of the scale length of the long strings γ vs $(a/a_0)^{\langle v^2\rangle-1}$ for the three code II runs from Fig. 10. $\langle v^2\rangle$ has been set to its scaling solution value $\langle v^2\rangle=0.43$. According to Kibble's one scale model, the curves should be straight lines which intersect at $(a/a_0)^{\langle v^2\rangle-1}=0$. The straight lines on the plot are least-square fits to the numerical results.

verging faster than this, and three of the straight lines fit to these curves do intersect each other at nearly the same point. But they do so at $(a/a_0)^{\langle v^2\rangle-1/2}=0.4$ rather than $(a/a_0)^{\langle v^2\rangle-1/2}=0$. The other two runs starting at intermediate densities do not seem to be converging to the same point, however. Part of the reason that the straight-line fits do not converge very well is that the curves have been compressed in the x direction because of the choice of the x-axis variable $(a/a_0)^{\langle v^2\rangle-1/2}$. This means that small variations in the γ values cause large changes in the slopes of the straight-line fits. A closer examination of the high- and low-density runs in Fig. 24 reveals that both the loop-production constant C and the mean-square velocity of the long strings $\langle v^2\rangle$ have some dependence on γ. Thus, the simplified one-scale model does not provide an accurate *quantitative* description of relaxation to the matter-era scaling solution.

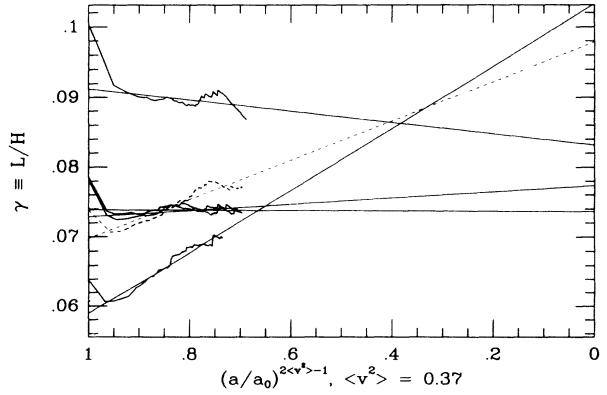

FIG. 24. The evolution of the scale length of the long strings γ vs $(a/a_0)^{\langle v^2\rangle-1/2}$ for the four code II matter-era runs from Fig. 19. $\langle v^2\rangle$ has been set to its scaling solution value $\langle v^2\rangle=0.37$. Kibble's one scale model implies that the curves should be straight lines which intersect at $(a/a_0)^{2\langle v^2\rangle-1}=0$.

Another test of Kibble's one-scale model is the dependence of the long-string scaling density on the intercommutation "probability" p. This refers to the probability that two strings will intercommute when they cross as opposed to passing through each other. This question can be answered in the context of classical field theory, so it depends only on the relative speed and angle with which the strings intersect. So far, the numerical simulations[21-23] of this process have failed to turn up a single case for which intercommutation does not occur. If they had, then we could include the velocity and angle dependence in our numerical calculations.

In his analytic model, Kibble included the possibility that strings might not intercommute every time they cross by inserting an intercommutation probability p multiplying the interaction terms $\dot\rho_{\text{to loops}}$ and $\dot\rho_{\text{from loops}}$ in Eq. (4.1). This just implies that for $p<1$, the rates for loop production and reconnection are reduced in proportion to p. In the simplified model, this has the effect of replacing the loop-production constant C by the product pC. This implies that the scaling solution value for $\gamma=\gamma_s$ should scale as p and that $\rho_{\text{LS}}=\mu/(\gamma H)^2\sim p^{-2}$. Figure 25 shows the radiation-era evolution of $\rho_{\text{LS}}H^2/\mu=1/\gamma^2$ for three runs of code I with p ranging from 1 to 0.5. These runs all started with the same initial conditions. Since $1/\gamma^2$ is dropping fairly rapidly at the end of each of these runs, we can conclude that γ_s cannot decrease faster than $p^{1/2}$ when p decreases. $\gamma_s\sim p^{1/4}$ is probably a better fit to our results, but we cannot exclude $\gamma_s\sim p^0$.

It may seem somewhat surprising that the Kibble-model prediction for the scaling of γ_s with p fails so dramatically whereas the simplified Kibble model does quite well in predicting the relaxation rate to the scaling solution in the radiation era. We believe that the reason that the simplified Kibble model does so well in predicting the relaxation to scaling is that the energy that is go-

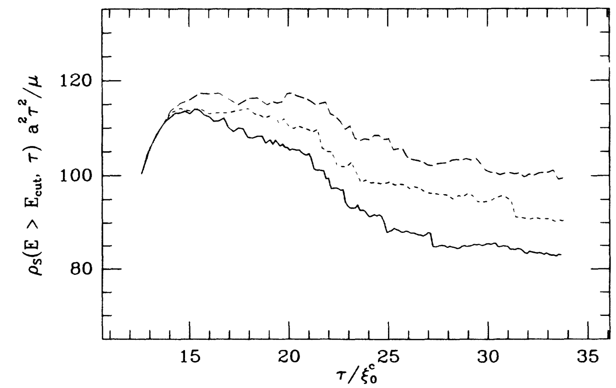

FIG. 25. The evolution of $\rho_{\text{LS}}H^2/\mu$ for runs of code I with values of the intercommutation probability $p=0.5$, 0.75, and 1 (from top to bottom respectively).

ing into the production of loops comes initially from wavelengths[28,30] on the string of order L, so that the time scale for loop production is of order L. The fact that the $p<1$ results do not fit the Kibble model probably means that for $p<1$, the features on the string destined to break off into loops get a second or third chance to break off the long-string network if they do not break off at the first crossing. This seems to imply that there should be a noticeable difference between the shape of long strings for a $p=1$ string network and a $p<1$ network.

B. The Kibble model in the transition era

In order to apply the simplified Kibble model [Eq. (4.4)] to the transition era, it is necessary to find some way to account for the variation of C and $\langle v^2\rangle$ between the radiation and matter eras. In their variation of the Kibble model, AT suggest the formula $\langle v^2\rangle=(1-L/R_H)/2$ which would imply that $\langle v^2\rangle_r=0.43$ and $\langle v^2\rangle_m=0.32$. This formula gives the correct qualitative behavior and accurately predicts the radiation-era value $\langle v^2\rangle_r=0.43\pm0.02$ but it does not do so well with the matter-era value $\langle v^2\rangle_m=0.37\pm0.02$. The AT velocity formula would give a 40% error in the exponent in Eq. (4.8). Since we are hoping for better accuracy than this, we have chosen to use the following formula which we have found to fit our transition-era runs very well:

$$\langle v^2\rangle(\tau)=\frac{\langle v^2\rangle_r+\langle v^2\rangle_m\tau/(7\tau_{\rm eq})}{1+\tau/(7\tau_{\rm eq})}=\frac{0.43+0.37\tau/(7\tau_{\rm eq})}{1+\tau/(7\tau_{\rm eq})}\ . \tag{4.9}$$

A comparison of this formula to the velocities measured in our transition-era simulations is given in Fig. 26.

It is considerably more difficult to measure the variation of C during the transition-era runs than it is to measure $\langle v^2\rangle$ because the value of C fluctuates quite a bit over the course of a run. We have found that the formula

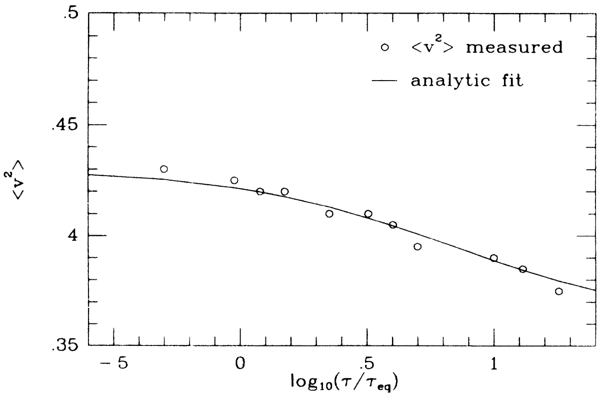

FIG. 26. The mean-square velocity of the long strings $\langle v^2\rangle$ is shown as a function of time as measured in our transition-era runs of code II. We estimate that the errors are ±0.01 statistical and ±0.01 systematic. (The systematic error would be about the same for all points.) The solid curve represents the analytic fit given in Eq. (4.9).

$$C(\tau)=\frac{C_r+C_m\tau/(10\tau_{\rm eq})}{1+\tau/(10\tau_{\rm eq})}=\frac{0.16+0.10\tau/(10\tau_{\rm eq})}{1+\tau/(10\tau_{\rm eq})} \tag{4.10}$$

is consistent with our measured variation of C, but it would not be fair to say it is a good fit because the measurements are so noisy.

If we insert the fit formulas (4.9) and (4.10) into the scale-length evolution equation (4.4), the resulting equation is linear and has the solution

$$L=\frac{\frac{1}{2}\int d\tau\, aC(\tau)\exp\left[-\int d\tau(\dot a/a)[1+\langle v^2\rangle(\tau)]\right]+\text{const}}{\exp\left[-\int d\tau(\dot a/a)[1+\langle v^2\rangle(\tau)]\right]}\ . \tag{4.11}$$

Despite our simple choices for $\langle v^2\rangle(\tau)$ and $C(\tau)$, the integrals in (4.11) must be done numerically. The result of this fit is shown as the dashed curve in Fig. 22. This curve seems to be a reasonably good fit to the general features of the runs shown in Fig. 22. It seems to fit the individual runs about as well as the runs which start at similar times match each other. Thus, once we have included the empirical fits for $\langle v^2\rangle(\tau)$ and $C(\tau)$, the Kibble model seems to describe the transition era fairly well.

V. CONCLUSIONS

In this paper, we have presented details of the two computer codes that we have written to study the evolution of a cosmic-string network in an expanding universe. With code I, we discovered that kinks play a very important role in string evolution, but we also found that the density of kinks grows so large that we were concerned that serious systematic errors related to our smoothing of these kinks might develop. This led us to develop a second, very-high-resolution code in which the smoothing of the kinks was kept to an absolute minimum.

We have presented the results of many numerical tests of these codes which have convinced us that both codes (and especially code II) are quite reliable. These tests show that our results are not influenced by the periodic boundary conditions, the density of sampling points, or the small loop cutoff (when it is made sufficiently small). The results of code II are found to agree extremely well with the highest-resolution results of code I as well as the results of the independent code of Allen and Shellard.[36] Our results do not agree so well with the results recently published by AT (Ref. 31), but we have shown that their code has excessive numerical smoothing that seems likely to explain the discrepancy. We have found that the radiation-era scaling solution value of the long-string energy density is $\rho_{\rm LS}=(52\pm10)\mu/H^2$ (where $H=a\int dt/a$ is the horizon size) while our matter-era value for the long-string density is $\rho_{\rm LS}=(31\pm7)\mu/H^2$.

We have compared our numerical results with an analytic model invented by Kibble. A simplified version of the Kibble model seems to fit our radiation-era results fairly well, and it gives a reasonable qualitative description of our matter-era results. The simplified Kibble model cannot account for our matter- and transition-era results quantitatively without some *ad hoc* modifications to account for the variation of the loop-production parameter C and the mean-square velocity of the strings $\langle v^2\rangle$ as a function of $\tau/\tau_{\rm eq}$ and/or γ. This contrasts with the simulations of AT (Ref. 31) who found their version of the Kibble model fit their results in almost every detail. This seems to be an indication that the physics missing from the Kibble model must come from scales that are small enough so that the AT code does not resolve them (i.e, $\lesssim L$).

The prevalence of small-scale structure on the strings is one of the most important new results of our high-resolution simulations. This small-scale structure is responsible for the fact that virtually no stable (non-self-intersecting) loops are produced by the string network except at the smallest scales that we can resolve. This significantly decreases the magnitude of the gravity-wave background expected to be generated by string decay,[14] and it means that cosmic-string loops will probably play only a minor role in seeding the formation of galaxies and large-scale structure.[29,30] A calculation of the pattern of structure formation to be expected in the string scenario is currently underway.

ACKNOWLEDGMENTS

The work of D.P.B. was supported in part by NSF Grant No. PHY80-19725 and by NASA Grant No. NAGW-765. Part of this research was done while F.R.B. was at the Theoretical Astrophysics Center at the University of California at Berkeley, and at the Institute of Geophysics and Planetary Physics, under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

APPENDIX A: NUMERICAL SMOOTHING IN STRING EVOLUTION CODES

In this appendix, we will study the effect of numerical smoothing in the numerical solution to Eqs. (2.12) and (2.13). We will consider these equations in flat spacetime ($\dot a=0$) because the analysis is much easier than the expanding case and because we can compare the numerical solutions with exact analytic solutions. Our flat-space analysis can be applied to expanding universe simulations because the smoothing due to the expansion is very much smaller than the numerical smoothing we are studying for waves with wavelengths much smaller than the horizon. (To apply our flat-space formulas to the expanding universe, we need only to replace $\Delta\sigma$ by $\epsilon\Delta\sigma$.) In the flat-space limit, (2.12) reduces to $s_\pm=\epsilon\sigma\pm\tau$, and (2.13) becomes simply $\dot{\mathbf{p}}=\dot{\mathbf{q}}=\dot\epsilon=0$. As described in the caption of Fig. 2 the numerical smoothing in the AT code is accomplished by interpolating the $\mathbf{p}$ and $\mathbf{q}$ functions so they can be stored at fixed values of σ (rather than fixed values of $s_\pm$) at every time step. The only complication is that because of the condition $\mathbf{p}^2=\mathbf{q}^2=1$, this interpolation must be done on a sphere. We can avoid having to impose this nonlinear constraint by considering a portion of string in which the $\mathbf{p}$ (or $\mathbf{q}$) function is confined to an arc of a great circle. We can also rid ourselves of this nonlinear constraint by considering only small amplitude or long-wavelength modes so that the patch of the sphere on which we do the interpolation is well approximated by a plane.

Since we have taken $\mathbf{p}$ to lie on a great circle, we can replace $\mathbf{p}$ by the variable ϕ which is just the angle variable of polar coordinates on the $\mathbf{p}$ plane. The difference equation describing the successive interpolations is just

$$\phi_j(t+\Delta t)=\alpha\phi_{j-1}(t)+(1-\alpha)\phi_j(t)\ , \tag{A1}$$

where $\alpha=\Delta t/\Delta\sigma$ is the ratio of the time step to the grid spacing and $\epsilon=1$. The subscript of ϕ_j refers to the jth spatial grid point. In order to generalize Eq. (A1) to the more realistic case where $\mathbf{p}$ deviates from a great circle, we must assume that $\mathbf{p}$ does not vary much between grid points as is the case when the amplitude of the short-wavelength modes is small. (Long-wavelength modes can still have large amplitudes.) In this case, we can approximate the small patch of the sphere that contains the grid points in question by a plane and use a two-dimensional analog of Eq. (A1). To apply the standard von Neumann stability analysis to Eq. (A1), we simply insert a Fourier expansion of $\phi(t,\sigma)$,

$$\phi(t,j\Delta\sigma)={\rm Re}\sum_k[\phi(t,k)e^{ikj\Delta\sigma}] \tag{A2}$$

into Eq. (A1). In the real solution, of course, $\phi(t,k)\equiv\phi(t_0,k)$, but with the difference equation (A1), we obtain

$$\frac{\phi(t+\Delta t,k)}{\phi(t,k)}=1-\alpha+\alpha e^{-ik\Delta\sigma}\ , \tag{A3}$$

or

$$\frac{|\phi(t+\Delta t,k)}{|\phi(t,k)|}=\sqrt{1-2\alpha(1-\alpha)(1-\cos k\Delta\sigma)}\ . \tag{A4}$$

For $k\Delta\sigma\ll1$ we obtain

$$\frac{|\phi(t_2,k)|}{|\phi(t_1,k)|}=e^{-(t_2-t_1)(1-\alpha)k^2\Delta\sigma/2}\ , \tag{A5}$$

for the decay of the amplitude of the wave during the time interval t_2-t_1.

In context of an expanding universe, it is useful to define the smoothing wavelength $\lambda_s=2\pi/k_s$ to be the wavelength for which $|\phi(t_2,k)/\phi(t_1,k)|=\frac{1}{2}$ when $t_2=2t_1$. Wavelengths smaller than λ_s will be numerically damped by more than a factor of 2 when t grows by a factor of 2 (or a grows by a factor of $\sqrt{2}$). Equation (A5) tells us that

$$\lambda_s = \pi \left[\frac{1-\alpha}{\ln 2} t \Delta\sigma \right]^{1/2} . \tag{A6}$$

If we had defined λ_s to be the wavelength of the wave whose amplitude decreases by a factor of 2 when $t_2 - t_1 = 0.1 t_2$, then λ_s would be reduced from the value given in Eq. (A6) by a factor of $\sqrt{5}$.

It is quite instructive to apply Eq. (A6) to the precise conditions in AT's largest radiation-era run.[31] This run started with $\Delta\sigma_0 / t_0 = \frac{1}{80}$, and the mean value of $\Delta\sigma$ remains roughly constant in physical units throughout the run. Their mean value for α decreases as $\alpha \propto a^{-1/2}$. Inserting these numbers into Eq. (A6), we obtain

$$\frac{\lambda_s}{2t} = 0.21 \frac{\sqrt{1-a^{-1/2}}}{a} . \tag{A7}$$

The function on the right-hand side of Eq. (A7) starts off at 0 for $a=1$, rises rapidly through 0.045 at $a=1.125$ to reach a maximum of 0.060 at $a=1.5625$. It then decreases very slowly reaching 0.048 at $a=2.8$ which corresponds to the end of AT's longest radiation-era run. Thus, throughout almost the entire run the smoothing wavelength λ_s remains within 15% of $0.053(2t)$ which is three quarters of AT's long-string scale length $L = 0.069(2t)$. We note that λ_s ranges from $11\Delta\sigma$ at the beginning of their run to $87\Delta\sigma$ at the end of the run. These are long enough wavelengths so that the long-wavelength approximation is reasonably accurate throughout the run, so our analysis is approximately correct even for modes that are not confined to a great circle on the Kibble-Turok sphere. Therefore, we conclude that the AT code will smooth very significantly any structure on a scale smaller than the long-string scale length $L = 0.07(2t)$.

APPENDIX B: THE STRETCHING OF KINKS BY THE EXPANSION

A simple formula for the decay of the kink amplitude with the expansion can be derived from Eq. (2.13). Let us consider a kink discontinuity in the left-moving wave with an amplitude given by

$$\mathbf{k} \equiv \mathbf{p}_+ - \mathbf{p}_- \quad \text{with } k \equiv |\mathbf{k}| , \tag{B1}$$

located at $s_+ = \overline{s}_+$. [$\mathbf{p}_\pm$ are just $\mathbf{p}(\overline{s}_+ \pm \delta]$ evaluated in the limit $\delta \rightarrow 0$.) Equation (2.13) implies that

$$\dot{\mathbf{k}} = \frac{\dot{a}}{a} [(\mathbf{p}_+ \cdot \mathbf{q}) \mathbf{p}_+ - (\mathbf{p}_- \cdot \mathbf{q}) \mathbf{p}_-] . \tag{B2}$$

An expression for $\dot{k}$ can be obtained by taking a scalar product with $\mathbf{k}$ and dividing by k^2 yielding

$$\frac{\dot{k}}{k} = \frac{\mathbf{k} \cdot \dot{\mathbf{k}}}{\mathbf{k}^2} = \frac{1}{2} \frac{\dot{a}}{a} [(\mathbf{q} \cdot \mathbf{p}_+) + (\mathbf{q} \cdot \mathbf{p}_-)] , \tag{B3}$$

where we have used the identity $\mathbf{p}_+^2 = \mathbf{p}_-^2 = 1$. Using Eq. (2.14), we can reduce this to

$$\frac{\dot{k}}{k} = -\frac{\dot{a}}{a} (1 - 2v^2) , \tag{B4}$$

where we have defined $v^2 \equiv (v_+^2 + v_-^2)/2$ to be the square velocity averaged over both sides of the kink. When we average over a long period of time Eq. (B4) indicates that the kink amplitude should decay as

$$k = k_0 \left[\frac{a}{a_0} \right]^{2\langle v^2 \rangle - 1} , \tag{B5}$$

where $\langle v^2 \rangle$ is the mean-square velocity of string containing the kink. For kinks on the long strings in the radiation era, we have $\langle v^2 \rangle \simeq 0.43$ which implies $k \propto a^{-0.14}$ so that the half-life of a radiation-era kink due to stretching is a factor of 140 in expansion. In the matter era, $\langle v^2 \rangle \simeq 0.37$ which implies $k \propto a^{-0.26}$ so that the half-life of a matter-era kink is a factor of 14 in expansion.

[1] T. W. B. Kibble, J. Phys. A **9**, 1387 (1976); Phys. Rep. **67**, 183 (1980).

[2] A. Vilenkin, Phys. Rep. **121**, 263 (1985), and references therein.

[3] Y. B. Zel'dovich, Mon. Not. R. Astron. Soc. **192**, 663 (1980).

[4] A. Vilenkin, Phys. Rev. Lett. **46**, 1169 (1980); **46**, 1496(E) (1980).

[5] E. Witten, Nucl. Phys. **B249**, 557 (1985).

[6] J. P. Ostriker, C. Thompson, and E. Witten, Phys. Lett. B **180**, 231 (1986).

[7] F. R. Bouchet, D. P. Bennett, and A. Stebbins, Nature (London) **335**, 410 (1988).

[8] A. Vilenkin, Phys. Rev. D **24**, 2082 (1981); Nature (London) **322**, 613 (1986).

[9] J. R. Gott, Astrophys. J. **288**, 422 (1985).

[10] C. Hogan and M. Rees, Nature (London) **311**, 109 (1984).

[11] A. Vilenkin, Phys. Lett. **107B**, 47 (1981).

[12] E. Witten, Phys. Rev. D **30**, 262 (1984).

[13] J. H. Taylor, Princeton University report, in the proceedings of the US-USSR Workshop on High Energy Astrophysics, Tbilisi, Moscow, 1989 (unpublished).

[14] F. R. Bouchet and D. P. Bennett, Phys. Rev. D **41**, 720 (1990).

[15] T. Vachaspati and A. Vilenkin, Phys. Rev. D **30**, 2036 (1984).

[16] A. E. Everett, Phys. Rev. D **24**, 858 (1981).

[17] Y. Nambu, in *Quarks and Symmetries,* proceedings of the International Conference on Symmetries and Quark Models, Detroit, Michigan, 1969, edited by R. Chand (Gordon and Breach, New York, 1970).

[18] T. W. B. Kibble, Nucl. Phys. **B252**, 227 (1985).

[19] N. Turok and P. Bhattacharjee, Phys. Rev. D **29**, 1557 (1984).

[20] A. Vilenkin, Phys. Rev. Lett. **46**, 1169 (1981).

[21] E. P. S. Shellard, Nucl. Phys. **B283**, 624 (1987).

[22] R. Matzner, Comput. Phys. **2**, 51 (1988).

[23] K. J. M. Moriarty, E. Myers, and C. Rebbi, Phys. Lett. B **207**, 411 (1988); Boston University Report Nos. BUHEP-88-26, 88-27, and 88-32 (unpublished).

[24] T. Vachaspati and A. Vilenkin, Phys. Rev. D **31**, 3052 (1985).

[25] D. P. Bennett, Phys. Rev. D **33**, 872 (1986).

[26] D. P. Bennett, Phys. Rev. D **34**, 3592 (1986).

[27] D. Mitchell and N. Turok, Phys. Rev. Lett. **58**, 1577 (1987); Nucl. Phys. **B294**, 1138 (1987).

[28] D. P. Bennett and F. R. Bouchet, Phys. Rev. Lett. **63**, 2776 (1989).

[29] F. R. Bouchet and D. P. Bennett, Astrophys. J. (to be published).

[30] F. R. Bouchet and D. P. Bennett, Princeton University Report No. PUPT-89-1138, 1989 (unpublished).

[31] A. Albrecht and N. Turok, Phys. Rev. D **40**, 973 (1989).

[32] A. Albrecht and N. Turok, Phys. Rev. Lett. **54**, 1868 (1985).

[33] D. P. Bennett and F. R. Bouchet, Phys. Rev. Lett. **60**, 257 (1988).

[34] F. R. Bouchet and D. P. Bennett, in *Cosmic Strings: The Current Status*, proceedings of the Yale Workshop, 1988, edited by F. S. Accetta and L. M. Krauss (World Scientific, Singapore, 1989).

[35] F. Accetta and L. Krauss, Phys. Lett. B **233**, 93 (1989).

[36] B. Allen and E. P. S. Shellard, Phys. Rev. Lett. **64**, 119 (1990).

[37] D. P. Bennett and F. R. Bouchet, Phys. Rev. Lett. **63**, 1334 (1989).

[38] T. W. B. Kibble and N. Turok, Phys. Lett. **116B**, 141 (1982).

[39] K. Maeda and N. Turok, Phys. Lett. B **202**, 376 (1988).

[40] R. Gregory, Phys. Lett. B **206**, 199 (1988).

[41] See, for example, R. L. Street, *The Analysis and Solution of Partial Differential Equations* (Brooks/Cole, Belmont, 1973), pp. 304–341.

[42] A. Albrecht and N. Turok, Princeton University Report No. PUPT-89-1133, 1989 (unpublished).

[43] A. Albrecht, Fermilab Report No. Fermilab-Conf-89/221A, 1989 (unpublished).

[44] N. Turok, Nucl. Phys. **B242**, 520 (1983).

[45] A. L. Chen, D. A. DiCarlo, and S. A. Hotes, Phys. Rev. D **37**, 863 (1988).

[46] R. J. Scherrer and W. H. Press, Phys. Rev. D **39**, 371 (1989).

[47] Here we follow the notation of Refs. 18, 25, and 26. The authors of Ref. 31 have defined γ to be the inverse of our γ.