CosmoTransitions: Computing cosmological phase transition temperatures and bubble profiles with multiple fields☆

Carroll L. Wainwright

Department of Physics, University of California, 1156 High St., Santa Cruz, CA 95064, USA

ARTICLE INFO

ABSTRACT

I present a numerical package (CosmoTransitions) for analyzing finite-temperature cosmological phase transitions driven by single or multiple scalar fields. The package analyzes the different vacua of a theory to determine their critical temperatures (where the vacuum energy levels are degenerate), their supercooling temperatures, and the bubble wall profiles which separate the phases and describe their tunneling dynamics. I introduce a new method of path deformation to find the profiles of both thin- and thick-walled bubbles. CosmoTransitions is freely available for public use.

Program summary

Program Title: CosmoTransitions
Catalogue identifier: AEML_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEML_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 8775
No. of bytes in distributed program, including test data, etc.: 621096
Distribution format: tar.gz
Programming language: Python.
Computer: Developed on a 2009 MacBook Pro. No computer-specific optimization was performed.
Operating system: Designed and tested on Mac OS X 10.6.8. Compatible with any OS with Python installed.
RAM: Approximately 50 MB, mostly for loading plotting packages.
Classification: 1.9, 11.1.
External routines: SciPy, NumPy, matplotLib
Nature of problem: I describe a program to analyze early-Universe finite-temperature phase transitions with multiple scalar fields. The goal is to analyze the phase structure of an input theory, determine the amount of supercooling at each phase transition, and find the bubble-wall profiles of the nucleated bubbles that drive the transitions.
Solution method: To find the bubble-wall profile, the program assumes that tunneling happens along a fixed path in field space. This reduces the equations of motion to one dimension, which can then be solved using the overshoot/undershoot method. The path iteratively deforms in the direction opposite the forces perpendicular to the path until the perpendicular forces vanish (or become very small). To find the phase structure, the program finds and integrates the change in a phase's minimum with respect to temperature.
Running time: Approximately 1 minute for full analysis of the two-scalar-field test model on a 2.5 GHz CPU.

1. Introduction

Phase transitions driven by scalar fields have probably played an important role in the very early evolution of the Universe. In most inflationary models, the dynamics are driven by the evolution of a scalar inflaton field, while at later times electroweak symmetry breaking is thought to be driven by a transition in the Higgs field vacuum expectation value. Electroweak scale physics is currently being probed by the LHC, so the phenomenology of the electroweak phase transition is of particular interest. A strongly first-order electroweak phase transition would have been a source of entropy production in the early Universe (thereby changing the evolution of its scale with respect to temperature) and produced a stochastic background of gravitational radiation [1], perhaps observable by
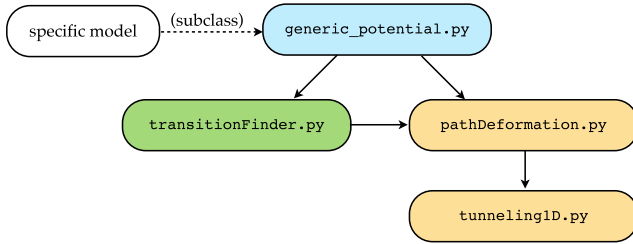
Fig. 1. Overview of the CosmoTransitions package file structure. The files `pathDeformation.py` and `tunneling1D.py` find critical bubble profiles, `transitionFinder.py` finds the minima of finite temperature potentials as a function of temperature and analyzes phase transitions, and `generic_potential.py` defines an abstract class that can easily be subclassed to examine a specific model.

future space-based gravitational radiation observatories [2]. In addition, a strongly first-order electroweak phase transition may have satisfied the Sakharov conditions [3] and been responsible for the current baryon asymmetry of the universe (for recent studies, see e.g. Refs. [4–18]), or may have affected the relic density of (for example) dark matter particles [19–21].

In the standard model, the electroweak phase transition is not strongly first-order unless the Higgs mass is below ~70 GeV [22–24], which is excluded by the current LEP bound of 114.4 GeV [25]. However, electroweak baryogenesis can be saved in extensions to the standard model, many of which include extra dynamic scalar fields (such as two-Higgs-doublet models [26–32]). The amount of produced baryon asymmetry depends crucially upon the dynamics of the phase transition, and particularly upon the bubble-wall profiles that separate the high- and low-temperature phases. These profiles are fairly easy to calculate using effective field theory if there is only one scalar field, but multiple fields greatly increase the computational complexity.

In this paper, I present an easy-to-use numerical package (CosmoTransitions) to analyze phase transitions in finite temperature field theory with multiple scalar fields. The program consists of three basic parts (see Fig. 1): modules for finding the tunneling solution (bubble wall profile) between different vacua, a module for finding critical temperatures and phase transitions, and an abstract class to define specific field-theoretic models and their effective potentials. In Section 2, I describe the algorithms for finding bubble wall profiles for both single and multiple fields. Section 3 describes the algorithm for finding phase transitions, while Section 4 describes how one can implement a specific model in a simple program. Finally, I present numerical results in Section 5 and conclude in Section 6.

To download the latest version of CosmoTransitions, visit http://chasm.ucsc.edu/cosmotransitions.

2. Calculating bubble profiles

First-order cosmological phase transitions proceed by the nucleation of bubbles of true vacuum out of metastable false vacuum states. The bubbles have both surface tension and internal pressure, so that large bubbles tend to expand and small bubbles tend to collapse. Critical bubbles—bubbles that are just large enough to avoid collapse—will drive the phase transition.

Given a Lagrangian

$$\mathcal{L} = \frac{1}{2}(\partial_\mu \vec{\phi})(\partial^\mu \vec{\phi}) - V(\vec{\phi}), \tag{1}$$

where $\vec{\phi}$ is a vector of scalar fields, a critical bubble can be found by extremizing the Euclidean action

$$S_E = \int d^d x \left[\frac{1}{2} \left(\partial_\mu \vec{\phi} \right)^2 + V(\vec{\phi}) \right], \tag{2}$$
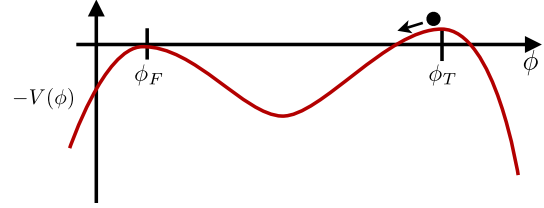

Fig. 2. The equations of motion for a field with a potential $V(\phi)$ can be thought of as the equations for a particle moving in an inverted potential $-V(\phi)$.

where $d = 4$ (3) for tunneling at zero (finite) temperature. This quantity is critical for finding the nucleation rate, and thereby determining the phase transition temperature and whether or not the transition actually happens (see Refs. [33–35] for seminal work on phase transitions in quantum field theory). The bubble nucleation rate per unit volume is $\Gamma/V = Ae^{-S_E}$ at zero temperature and $\Gamma/V = Ae^{-S_E/T}$ at finite temperature. The prefactor A is quite difficult to calculate, but it has only weak temperature dependence and can generally be estimated on dimensional grounds (see, e.g., Ref. [36]). By requiring that the expectation value for one bubble to nucleate per Hubble volume is $\sim \mathcal{O}(1)$, one can show that the bubble nucleation temperature for weak-scale fields is given by $S_E/T \sim 140$.

Assuming spherical symmetry, the bubble's equations of motion are

$$\frac{d^2 \vec{\phi}}{d\rho^2} + \frac{\alpha}{\rho} \frac{d\vec{\phi}}{d\rho} = \nabla V(\vec{\phi}). \tag{3}$$

At finite temperature, ρ is simply the spatial radial coordinate and $\alpha = 2$. At zero temperature, $\rho^2 = r^2 - t^2$ and $\alpha = 3$. The program defaults to $\alpha = 2$, but this can be overridden when using either `pathDeformation.py` or `tunneling1D.py` directly, ignoring the temperature-dependent routines in `transitionFinder.py`. Let $\vec{\phi}_T$ and $\vec{\phi}_F$ denote the true and false vacua, respectively. Then in order for the solution to match the field at infinity, we require that $\vec{\phi}(\infty) = \vec{\phi}_F$. We also demand that $\left. \frac{d\vec{\phi}}{dr} \right|_{\rho=0} = 0$ so that the bubble is non-singular at the origin.

Note that this procedure is limited by the validity of perturbation theory and of the effective potential. Ideally, once all model parameters are set, one should perform a lattice Monte-Carlo simulation to compute the nucleation rate and bubble profile as accurately as possible [37,38]. However, this is often computationally impractical.

2.1. One-dimensional solution

When the field has only one dimension, the solution to the critical bubble profile can easily be solved by the overshoot/undershoot method (see, e.g., Ref. [39]). Here, it is easiest to think of the problem as that of a classical particle moving under the influence of the inverted potential $-V(\phi)$ plus a peculiar looking friction term, where ϕ takes on the role of a spatial coordinate and ρ acts as the time coordinate. The problem then is to find the initial placement of the particle near ϕ_T such that it rolls down the potential and comes to a stop at ϕ_F when $\rho = \infty$ (see Fig. 2). If the particle rolls past (overshoots) ϕ_F, then the initial placement was too close to ϕ_T. If it does not have enough energy to make it to ϕ_F (an undershoot), then the initial placement needs to be closer to $\vec{\phi}_T$. Through trial and error, one can find the initial placement to arbitrary precision.

The code presented with this paper follows the general overshoot/undershoot method, implemented by the class `tunneling1D.bubbleProfile`. Calculation of thick-walled bubbles is straightforward, but thin-walled bubbles require extra consideration. In thin-walled bubbles, the transition from $\phi \approx \phi_T$ to $\phi \approx \phi_F$
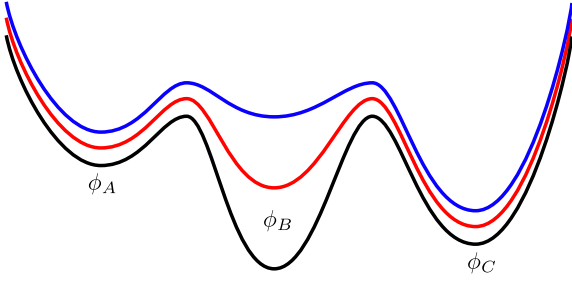
Fig. 3. A problematic potential with more than two minima. A tunneling solution from ϕ_A to ϕ_C is only guaranteed to exist for the topmost potential (blue line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 4. Path deformation in two field dimensions. The normal force exerted on the starting path (blue straight line) pushes it in the direction of the true tunneling solution (green curved line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

happens over a distance much shorter than the bubble's overall size. In the particle analogy, the particle sits very close to ϕ_T for a very long time before quickly rolling down the potential and stopping at ϕ_F. Extreme accuracy in $\phi_0 = \phi(\rho = 0)$ would be needed to reliably calculate the wall profile, since a small change in ϕ_0 would lead to a large change in the radius of the bubble. Instead, I define a new variable x such that $\phi_0 = \phi_T + e^{-x}(\phi_F - \phi_T)$, and use it as the initial condition to vary instead.

For small ρ and $\phi \approx \phi_T$, the equation of motion can be approximated as

$$\frac{d^2\phi}{d\rho^2} + \frac{\alpha}{\rho}\frac{d\phi}{d\rho} = \frac{d^2V}{d\phi^2}\bigg|_{\phi=\phi_T}(\phi - \phi_T) \qquad (4)$$

which has the exact solution

$$\phi(\rho) - \phi_T = C\rho^{-\nu}I_\nu\left(\rho\sqrt{b}\right) \qquad (5)$$

where $\nu = \frac{\alpha-1}{2}$, $b = \frac{d^2V}{d\phi^2}\big|_{\phi=\phi_T}$, and I_ν is the modified Bessel function of the first kind. For $\alpha = 2$, this simplifies to $\phi(\rho) - \phi_T \propto \frac{1}{\rho}\sinh(\rho\sqrt{b})$. These can be numerically inverted to find $\rho(\phi)$, which allows one to calculate the approximate radius of a thin-walled bubble without performing any integration. The integration can then start at the edge of the bubble wall, which both increases accuracy and decreases computation time.

The overshoot/undershoot implementation is fairly robust, but there are situations in which it either does not work or is unreliable. Consider the potentials shown in Fig. 3. They each have three minima $\phi_A < \phi_B < \phi_C$, with $V(\phi_A) > V(\phi_C)$. If one tries to calculate the bubble profile for tunneling from ϕ_A to ϕ_C, only the topmost potential is unproblematic. There, there will be a solution where the field starts near ϕ_C, rolls over the bump at ϕ_B, and ends up at ϕ_A. In the bottommost potential, no such solution exists. The minimum at ϕ_B is the true vacuum of the theory, so the field will tunnel from ϕ_A to ϕ_B, ignoring ϕ_C completely. The overshoot/undershoot method will return this solution (any initial $\phi_0 > \phi_B$ will register as an undershoot), but it will not perform any optimization for thin-walled bubbles and may have poor accuracy. In the middle case, with $V(\phi_A) > V(\phi_B) > V(\phi_C)$, there may or may not be a direct tunneling solution from ϕ_A to ϕ_C. Using the particle analogy, it could be that any particle that starts near ϕ_C and has enough momentum to get beyond ϕ_B will necessarily overshoot ϕ_A. In this case, the field can only tunnel to ϕ_C in two steps: first by tunneling to ϕ_B, and then by tunneling to ϕ_C in a separate nucleation event. The algorithm in `tunneling1D.bubbleProfile` will only ever return the first of these transitions, and it may have poor accuracy.
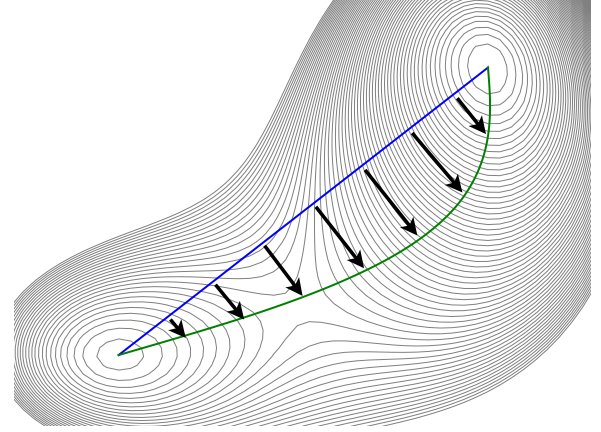
2.2. Multi-dimensional solution and path deformation

With the solution to the one-dimensional problem in hand, we are ready to tackle the more challenging problem with multiple field dimensions. The overshoot/undershoot method no longer works because we required the unique topology of the one-dimensional case to determine whether a particular solution overshot or undershot the boundary condition at $\rho = \infty$. Instead, I propose a method of path deformation.

First, assume as an initial guess that the tunneling occurs on some fixed path in field space. That is, $\vec{\phi}_{guess} = \vec{\phi}(x)$, where x parametrizes the path and for simplicity we require that $\left|\frac{d\vec{\phi}}{dx}\right| = 1$. The equations of motion split into two parts—one parallel and one perpendicular to the path:

$$\frac{d^2x}{d\rho^2} + \frac{\alpha}{\rho}\frac{dx}{d\rho} = \frac{\partial}{\partial x}V[\vec{\phi}(x)] \qquad (6)$$

$$\frac{d^2\vec{\phi}}{dx^2}\left(\frac{dx}{d\rho}\right)^2 = \nabla_\perp V(\vec{\phi}), \qquad (7)$$

where $\nabla_\perp V$ represents the components of the gradient of V that are perpendicular to the path. Eq. (6) is the same as the one-dimensional equation of motion, which we can solve by the overshoot/undershoot method. If the path guess $\vec{\phi}(x)$ is correct, then the solution to Eq. (6) will also solve Eq. (7). The trick then is to find the right path. A similar approach of path deformation has been proposed by Ref. [40], while Refs. [41–43] use alternate methods to solve the equations of motion. I have previously employed the basic algorithm described below in the context of zero-temperature phase temperatures and singlet scalar dark matter models [44].

Using the particle analogy, we can think of the path guess as a fixed track on which the particle moves through the multidimensional space. Eq. (6) describes the forces parallel to the track and thus determine the particle's speed. Eq. (7) does not effect the motion of the particle, but instead determines the normal force N that the track must exert on the particle to keep it from falling off: $N = \frac{d^2\vec{\phi}}{dx^2}\left(\frac{dx}{d\rho}\right)^2 - \nabla_\perp V(\vec{\phi})$. For the right path, $N = 0$. Given a starting guess, one can deform the path to the correct solution by continually pushing it in the direction of N (see Fig. 4).

I implement this general method in the CosmoTransitions package using B-splines in the module `pathDeformation.py`. Each path is written as a linear combination of spline basis functions
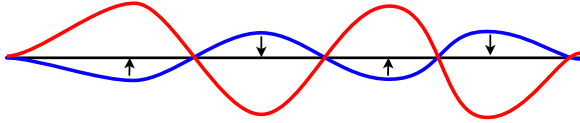
Fig. 5. Example of error correction in deformation. The normal force will push wiggles down towards the straight line solution. But if the stepsize is too large, the wiggles will get reversed and amplified instead.

plus a linear component connecting its ends: $\vec{\phi}(y) = \sum_i \beta_i \vec{\phi}_i(y) + (\vec{\phi}_0 - \vec{\phi}_F)y + \vec{\phi}_F$, where y parametrizes the path ($0 \leq y \leq 1$, and generally $\left| \frac{d\vec{\phi}}{dy} \right| \neq 1$), β_i are the spline coefficients, $\vec{\phi}_i(y)$ are the basis functions, and $\vec{\phi}_i(0) = \vec{\phi}_i(1) = 0$. This fixes the path's endpoints at $\vec{\phi}_0 \sim \vec{\phi}(\rho = 0)$ and $\vec{\phi}_F$. Generally, only a small number of basis functions is needed to accurately model the path ($\sim$10 per field direction) unless it contains sharp bends or many different curves.

Before any deformation, the algorithm first calculates the bubble profile along the starting path using the overshoot/undershoot method. Then it deforms the path in a series of steps without recalculating either the one-dimensional profile or $\left| \frac{d\vec{\phi}}{d\rho} \right|$. At each step it calculates the normal force for a relatively large number of points ($\sim$100) along the path, rescales the normal force by $|\vec{\phi}_T - \vec{\phi}_F|/|\nabla V|_{\max}$, and moves the points in that direction times some small stepsize. If the one-dimensional solution is thick-walled (that is, $\vec{\phi}(\rho = 0)$ is not very close $\vec{\phi}_T$), the algorithm also moves $\vec{\phi}_0$ in the direction of N averaged over the first several points (note that $N(\rho = 0) = 0$ as long as the path aligns with ∇V). Otherwise, $\vec{\phi}_0$ stays fixed at $\vec{\phi}_T$. It then recalculates the spline coefficients by a least-squares fit to the moved points with the restriction that the path aligns with ∇V at $\vec{\phi}_0$ when thick-walled. The deformation converges when the normal force is much smaller than forces parallel to the path. At this point, the algorithm recalculates $\left| \frac{d\vec{\phi}}{d\rho} \right|$ and deforms the path again. After repeating this procedure a few times, the deformation should converge after a single step and the algorithm will return the final tunneling solution.

Choosing an appropriate stepsize is important. Errors in individual steps are generally self-correcting, but only for small stepsizes. Consider a situation in which the correct tunneling path is a straight line, but we introduce errors in the deformation to add wiggles (see Fig. 5). A small stepsize in deformation will tend to smooth out the errors, but a large stepsize will instead reverse and amplify them. By checking for such reversals, the algorithm can keep the stepsize at the appropriate level.

3. Exploring phase structures

In order to determine the characteristics of a phase transition, we must first find where, and at what temperatures, the various phases exist. In theories with spontaneously broken symmetries there is at least one zero-temperature phase and there is generally one high-temperature symmetry-restoring phase. If these phases coexistence at some temperature, then there is likely a first-order phase transition between them. However, even in relatively simple models there can be intermediate phases (see, e.g., Ref. [45]) which can lead to secondary phase transitions, or can change the quality of the primary transition. Therefore, it is helpful to find the location of the minima as a function of temperature.

By writing the potential about any point $\vec{\phi}'$ as

$$V(\vec{\phi}, T) = a_i + b_i(\phi_i - \phi_i') + \frac{1}{2}M_{ij}(\phi_i - \phi_i')(\phi_j - \phi_j') + \cdots, \quad (8)$$

where $b_i = \frac{\partial V}{\partial \phi_i}$ and $M_{ij} = \frac{\partial^2 V}{\partial \phi_i \partial \phi_j}$, one can show that there is a (nearby) minimum at $\vec{\phi}_{\min} - \vec{\phi}' = -M^{-1}\vec{b}$. Therefore, the change in the minimum with respect to temperature is

$$\frac{\partial \vec{\phi}_{\min}}{\partial T} = -M^{-1}\frac{\partial \vec{b}}{\partial T} - \left(\frac{\partial}{\partial T}M^{-1} \right)\vec{b} = -M^{-1}\frac{\partial \vec{b}}{\partial T}, \quad (9)$$

since $\vec{b} = 0$ at the minimum. Given a minimum at a single temperature, this allows one to find all of the minima of a given phase as a function of T. A singular matrix M indicates a rapid change in the minimum caused by either the disappearance of the phase or a second-order phase transition. A negative eigenvalue in M indicates that the extremum is no longer a minimum.

Practically, it is much easier to use this equation in conjunction with a minimization routine than it is to integrate it by itself. The algorithm in `transitionFinder.traceMinimum` uses the Nelder–Mead downhill simplex method [46] to find the local minimum at particular temperatures, and then uses Eq. (9) to find how the minimum changes. The former acts as an error check on the latter, which allows for an adaptive stepsize in the temperature. Once the phase disappears, the downhill simplex method can search for a new phase and then trace that. In this manner, one can trace the phase structure of the entire theory (assuming that each phase can be found by minimizing the potential at one of the ends of the other phases using the downhill simplex method). If there are multiple distinct phases at zero temperature, it is necessary to manually enter their (approximate) locations in order to trace them independently.

4. Structure of a simple program

There are essentially three parts to a simple program using my code: the tunneling algorithms and phase tracing algorithms described above, and the implementation of a specific model. Much of this last task happens in the `generic_potential` class, which must be subclassed to study any particular theory.

From the point of view of the finite temperature effective potential, the theory is completely determined by the tree-level potential and field-dependent mass spectrum. The `generic_potential` class calculates the one-loop corrections from these masses using $\overline{MS}$ renormalization [47,48]

$$V_1(\vec{\phi}) = \pm \frac{1}{64\pi^2} \sum_i n_i m_i^4(\vec{\phi}) \left[\log \frac{m_i^2(\vec{\phi})}{Q^2} - c_i \right], \quad (10)$$

where n_i and m_i are the numbers of degrees of freedom and the field-dependent masses of each particle species. The quantity Q is the renormalization scale; $c = 1/2$ for gauge boson transverse modes and $3/2$ for all other particles; and the upper (lower) sign is for bosons (fermions). The one-loop finite-temperature corrections are

$$V_1(\vec{\phi}, T) = \frac{T^4}{2\pi^2} \sum_i n_i J_{\mp} \left[\frac{m_i(\vec{\phi})}{T} \right] \quad (11)$$

with

$$J_{\mp}(x) = \pm \int_0^\infty dy\, y^2 \log \left(1 \mp e^{-\sqrt{y^2 + x^2}} \right). \quad (12)$$

The functions $J_{\mp}(x)$ are implemented in the module `finiteT.py` using direct integration and cubic interpolation. Of course, one can add additional structure (such as counter-terms) to a subclass.

To create a fully functioning model using the CosmoTransitions package, one need only subclass `generic_potential` and overwrite four functions: the initialization function `init()` to
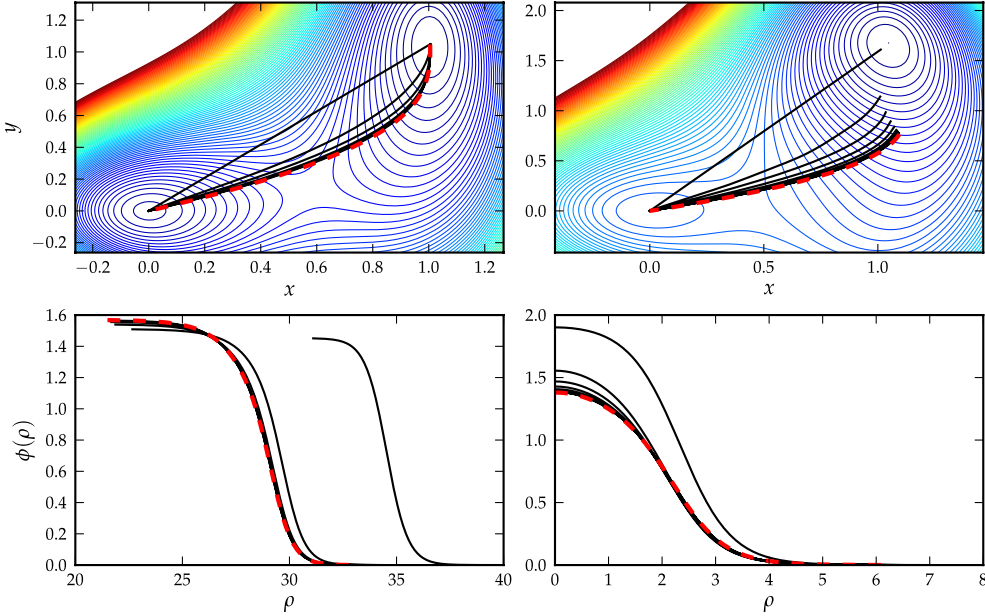
Fig. 6. Top: the potential $V(x, y)$ given by Eq. (13) for $\delta = 0.02$ (left) and $\delta = 0.4$ (right). Black lines show successive deformations of the tunneling path, while red dashed lines show direct integration of Eq. (3) with manually chosen initial conditions. Note that the x and y axes are not to scale, so ∇V does not appear perpendicular to the contour lines. Bottom: the bubble profiles associated with each of the above deformations, where $\phi(\rho)$ is measured along the path. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

specify the number of field dimensions, the tree-level potential function V0(), and the mass-spectrum functions boson_massSq() and fermion_massSq().

At high temperature, the perturbative expansion of the effective potential fails and must be supplemented with daisy resummation, which amounts to adding thermal masses to all bosonic propagators [49,50]. These temperature-dependent masses can easily be added directly to boson_massSq(). Generally, one only needs to add the thermal masses to zero-mode propagators, corresponding to the cubic terms in $J(x)$, but it is easier in this program to add them to all propagators and all terms in $J(x)$.

Additionally, one should overwrite approxZeroTMin() to return the approximate locations of the zero temperature minima, especially if more than one minimum exists. Calling the function getPhases() will run routines from transitionFinder.py and calculate the phase structure of the theory. The critical temperature(s) (the temperature of degenerate minima between two phases) can be found by calling the function calcTcTrans(), while the function calcFullTrans() will find the amount of supercooling and the critical bubble profile for each transition. The supercooling criterion defaults to $S_E/T = 140$, which corresponds to the nucleation temperature in the electroweak phase transition, but this can easily be changed to study other phase transitions using the parameter nuclCriterion in calcFullTrans().

5. Numerical results

5.1. Deformation

To test the path deformation and tunneling routines, I use a simple potential given by

$$V(x, y) = \left(x^2 + y^2\right)\left[1.8(x - 1)^2 + 0.2(y - 1)^2 - \delta\right]. \quad (13)$$

This has one local minimum at $x = y = 0$, and a global minimum near $x = y = 1$. For $\delta \ll 1$, the phases are nearly degenerate and any tunneling between them will be thin-walled.

I run the pathDeformation.fullTunneling class for both thin ($\delta = 0.02$) and thick-walled ($\delta = 0.4$) potentials, with results

shown in Fig. 6. Each line represents 15 individual deformation steps with adaptive stepsizes on the order of 0.005. In this case, and in general, the thick-walled case converges more slowly due to the added complication of moving $\phi_0 = \phi(\rho = 0)$ with each step. The thin-walled case converges in ~ 60 steps, while the thick-walled case takes ~ 150 steps.

To check the deformation solution, I numerically integrate the equations of motion (3), using manual trial and error to find the correct initial conditions. These are shown in Fig. 6 as the red dashed lines. The deformation algorithm gets extremely close to— but not exactly to—the integrated solution. In the thin-walled case, the deformed path is within 0.1% of the integrated solution (defined as the largest distance separating the two paths divided by the total path length), while in the thick-walled case the error is 0.3%.

5.2. Calculating transition temperatures

To demonstrate the transition finding algorithms, I consider a theory with two scalar fields and a tree-level potential

$$V_0(s_1, s_2) = \frac{1}{8}\frac{m_1^2}{v^2}\left(s_1^2 - v^2\right)^2 + \frac{1}{8}\frac{m_2^2}{v^2}\left(s_2^2 - v^2\right)^2 - \mu^2 s_1 s_2. \quad (14)$$

When $\mu^2 = 0$, the theory has four identical minima located at $(s_1, s_2) = (\pm v, \pm v)$ and $(s_1, s_2) = (\pm v, \mp v)$. The effect of $\mu^2 > 0$ is to lower the minima at $(\pm v, \pm v)$ and raise those at $(\pm v, \mp v)$. The tree-level scalar mass-matrix is

$$m_{ij}^2(s_1, s_2) = \frac{1}{2v^2}\begin{pmatrix} m_1^2(3s_1^2 - v^2) & -\mu^2 \\ -\mu^2 & m_2^2(3s_2^2 - v^2) \end{pmatrix}, \quad (15)$$

so that the masses at the tree-level minima are m_1 and m_2 when $\mu^2 = 0$. I also add an extra bosonic degree of freedom with the field-dependent mass

$$m_X(s_1, s_2) = y_A^2\left(s_1^2 + s_2^2\right) + y_B^2 s_1 s_2. \quad (16)$$

The peculiar coupling y_B is designed to lift the minima at $(\pm v, \pm v)$ relative to $(\pm v, \mp v)$ at finite temperature so that there may be a phase transition between the two.

Table 1
Model parameters.

m_1	m_2	μ	v	Q	y_A^2	y_B^2	n_X
120 GeV	50 GeV	25 GeV	246 GeV	246 GeV	0.10	0.15	30
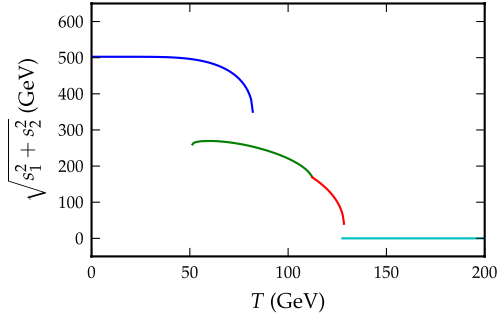
**Fig. 7.** The position of the minima as a function of temperature.

I examine a model with the parameters given in Table 1, where Q is the renormalization scale used in Eq. (10) and n_X is the number of degrees of freedom assigned to the extra boson with mass m_X. For this particular choice of parameters there are four local minima at tree-level, but loop corrections destroy the zero-temperature minima at $(\pm v, \mp v)$. I chose the parameters primarily to showcase multiple transitions within a single model, not for any physically motivated purpose. The tunneling condition is $S_E/T = 140$.

Fig. 7 shows the phase structure of the model, and Fig. 8 shows the general evolution of the potential as a function of temperature. At high temperatures, there is a single phase with $s_1 = s_2 = 0$. When the temperature drops to 128.2 GeV, a new phase appears at $(s_1, s_2) = (43, -33)$ GeV. The system tunnels to this phase at $T = 128.1$ GeV via a first-order phase transition (see Fig. 9), and the high-temperature phase disappears by $T = 127.6$ GeV. At $T = 112.2$ GeV, there is a small discontinuity due to the non-

analyticity of $J_-(m/T)$ when $m = 0$. At the default resolution the algorithm registers this as a second-order phase transition, but for a small range of temperatures (112.14–112.22 GeV) there are actually two distinct phases separated by $\Delta\phi \approx 2$ GeV (that is, the algorithm sees that each phase terminates near $T = 112.2$ GeV, but it does not probe the small temperature window where they coexist. Instead, it assumes that they are continuously linked by a second-order transition). Immediately below this the system can be thought of as being in the phase associated with the tree-level vev at $(s_1, s_2) = (+v, -v)$. As the system cools, the phase associated with $(s_1, s_2) = (+v, +v)$ appears, and by $T = 75.2$ GeV the two phases are degenerate. Below this the former phase is metastable, but it does not transition until $T = 54.4$ GeV, which is just above the point at which it disappears. Thus, the final tunneling is very thick-walled (see Fig. 9).

In order to reproduce these calculations, one needs to create an instance of the class `testModels.model1` with parameters given by Table 1 and call the class functions `getPhases()`, `calcTcTrans()` and `calcFullTrans()` with their default parameters. Much of the plotting can be handled by the functions `plot2d()` and `plotPhasesPhi()`.

To check that the code works with three scalar fields, I simply add an extra field to the potential such that its minimum is always zero: $V_0(s_1, s_2, s_3) = V_0(s_1, s_2) + v^2 s_3^2$. I then rotate s_2 and s_3 by 45°. This gives an easily visualizable potential with non-trivial minima. Running the above commands produces the same results as in the two-dimensional model within the default error tolerances. I successfully tested a four dimensional model in a similar way. In four dimensions, the transition finding routine with default resolution labels the highest temperature transition as second-order instead of weakly first-order, but this can be corrected by
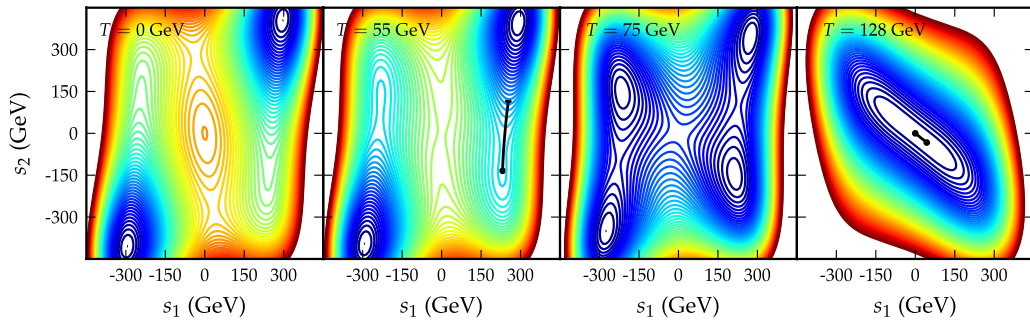
Fig. 8. The finite-temperature effective potential $V(s_1, s_2, T)$ at $T = 0, 55, 75$, and 128 GeV. Thick black lines at $T = 55$ and 128 GeV show the tunneling paths for the phase transitions.
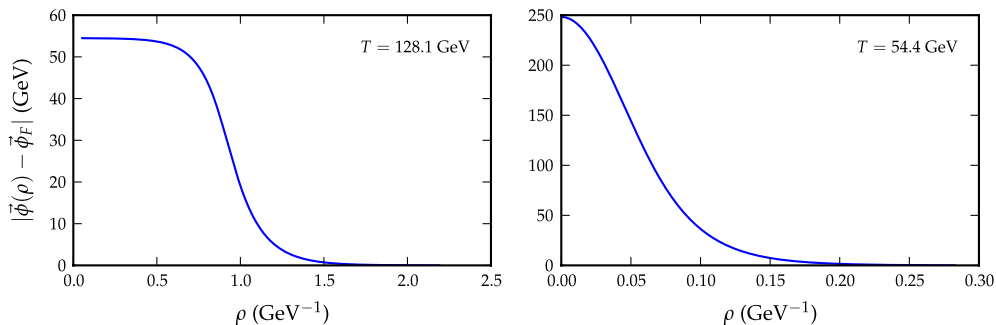
Fig. 9. Bubble profiles along the tunneling paths for both the high- and low-temperature phase transitions. The high-temperature profile is (relatively) thin-walled, while the low-temperature profile is very much thick-walled.

increasing the resolution. Higher dimensional models have not been tested, but the code was written to support an arbitrary number of scalar fields.

6. Conclusion

I presented the publicly available CosmoTransitions package to analyze cosmological phase transitions. This included algorithms to find the temperature-dependent phase minima, their critical temperatures, and the actual nucleation temperatures and tunneling profiles of the transitions. I introduced a novel method of path deformation to find the profiles, which I then demonstrated in simple test cases to accuracies of order $\sim 0.1\%$. The deformation algorithm has been successfully tested in 2 and 3 dimensions with both thick- and thin-walled profiles, but it should work in any number of higher dimensions as well.

CosmoTransitions is designed to be easily extensible, with minimal work needed on the part of the model builder. A new model can be created by subclassing the `generic_potential` class and specifying only the number of dynamic scalar fields, the tree-level potential, and the field-dependent particle spectrum. This will (hopefully) allow for the quick analysis of phase transitions in many extensions to the standard model.

Acknowledgments

References

[1] E. Witten, Cosmic separation of phases, Phys. Rev. D 30 (1984) 272–285. http://dx.doi.org/10.1103/PhysRevD.30.272.

[2] R. Blanford, et al., New Worlds, New Horizons in Astronomy and Astrophysics, National Academies Press, Washington, DC, 2010.

[3] A. Sakharov, Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe, Pisma Zh. Eksp. Teor. Fiz. 5 (1967) 32–35. Reprinted in Kolb, E.W. Turner, M.S. (Eds.): The early universe pp. 371–373, and in Lindley, D. et al. (Eds.): Cosmology and Particle Physics pp. 106–109, and in Sov. Phys. Usp., 34 (1991), 392–393 [Usp. Fiz. Nauk, 161 (5) (1991), 61–64].

[4] M.S. Carena, M. Quiros, C. Wagner, Opening the window for electroweak baryogenesis, Phys. Lett. B 380 (1996) 81–91. http://dx.doi.org/10.1016/0370-2693(96)00475-3. arXiv:hep-ph/9603420.

[5] S. Huber, P. John, M. Schmidt, Bubble walls, CP violation and electroweak baryogenesis in the MSSM, Eur. Phys. J. C 20 (2001) 695–711. arXiv:hep-ph/0101249.

[6] J. Kang, P. Langacker, T.-J. Li, T. Liu, Electroweak baryogenesis in a supersymmetric U(1)-prime model, Phys. Rev. Lett. 94 (2005) 061801. http://dx.doi.org/10.1103/PhysRevLett.94.061801. arXiv:hep-ph/0402086.

[7] M. Carena, G. Nardini, M. Quiros, C. Wagner, The baryogenesis window in the MSSM, Nuclear Phys. B 812 (2009) 243–263. http://dx.doi.org/10.1016/j.nuclphysb.2008.12.014. arXiv:0809.3760.

[8] K. Funakubo, E. Senaha, Electroweak phase transition, critical bubbles and sphaleron decoupling condition in the MSSM, Phys. Rev. D 79 (2009) 115024. http://dx.doi.org/10.1103/PhysRevD.79.115024. arXiv:0905.2022.

[9] C.-W. Chiang, E. Senaha, Electroweak phase transitions in the secluded U(1)-prime-extended MSSM, J. High Energy Phys. 1006 (2010) 030. http://dx.doi.org/10.1007/JHEP06(2010)030. arXiv:0912.5069.

[10] A. Ahriche, S. Nasri, Electroweak phase transition in the U(1)'-MSSM, Phys. Rev. D 83 (2011) 045032. http://dx.doi.org/10.1103/PhysRevD.83.045032. arXiv:1008.3106.

[11] D.J. Chung, A.J. Long, Electroweak phase transition in the munuSSM, Phys. Rev. D 81 (2010) 123531. http://dx.doi.org/10.1103/PhysRevD.81.123531. arXiv:1004.0942.

[12] S. Profumo, M.J. Ramsey-Musolf, G. Shaughnessy, Singlet Higgs phenomenology and the electroweak phase transition, J. High Energy Phys. 0708 (2007) 010. http://dx.doi.org/10.1088/1126-6708/2007/08/010. arXiv:0705.2425.

[13] A. Ahriche, What is the criterion for a strong first order electroweak phase transition in singlet models? Phys. Rev. D 75 (2007) 083522. http://dx.doi.org/10.1103/PhysRevD.75.083522. arXiv:hep-ph/0701192.

[14] S. Ham, S.-A. Shim, S. Oh, Electroweak phase transition in an extension of the standard model with scalar color octet, Phys. Rev. D 81 (2010) 055015. http://dx.doi.org/10.1103/PhysRevD.81.055015. arXiv:1002.0237.

[15] C. Lee, V. Cirigliano, M.J. Ramsey-Musolf, Resonant relaxation in electroweak baryogenesis, Phys. Rev. D 71 (2005) 075010. http://dx.doi.org/10.1103/PhysRevD.71.075010. arXiv:hep-ph/0412354.

[16] D.J. Chung, B. Garbrecht, M.J. Ramsey-Musolf, S. Tulin, Yukawa interactions and supersymmetric electroweak baryogenesis, Phys. Rev. Lett. 102 (2009) 061301. http://dx.doi.org/10.1103/PhysRevLett.102.061301. arXiv:0808.1144.

[17] V. Cirigliano, Y. Li, S. Profumo, M.J. Ramsey-Musolf, MSSM baryogenesis and electric dipole moments: an update on the phenomenology, J. High Energy Phys. 1001 (2010) 002. http://dx.doi.org/10.1007/JHEP01(2010)002. arXiv:0910.4589.

[18] D.J. Chung, B. Garbrecht, M. Ramsey-Musolf, S. Tulin, Supergauge interactions and electroweak baryogenesis, J. High Energy Phys. 0912 (2009) 067. http://dx.doi.org/10.1088/1126-6708/2009/12/067. arXiv:0908.2187.

[19] C. Wainwright, S. Profumo, The impact of a strongly first-order phase transition on the abundance of thermal relics, Phys. Rev. D 80 (2009) 103517. http://dx.doi.org/10.1103/PhysRevD.80.103517. arXiv:0909.1317.

[20] D. Chung, A. Long, L.-T. Wang, Probing the cosmological constant and phase transitions with dark matter, Phys. Rev. D 84 (2011) 043523. http://dx.doi.org/10.1103/PhysRevD.84.043523. arXiv:1104.5034.

[21] D.J.H. Chung, A.J. Long, Cosmological constant, dark matter, and electroweak phase transition, Phys. Rev. D84 (2011) 103513. arXiv:1108.5193.

[22] K. Kajantie, M. Laine, K. Rummukainen, M.E. Shaposhnikov, Is there a hot electroweak phase transition at m(H) larger or equal to m(W)? Phys. Rev. Lett. 77 (1996) 2887–2890. http://dx.doi.org/10.1103/PhysRevLett.77.2887. arXiv:hep-ph/9605288.

[23] F. Csikor, Z. Fodor, J. Heitger, Endpoint of the hot electroweak phase transition, Phys. Rev. Lett. 82 (1999) 21–24. http://dx.doi.org/10.1103/PhysRevLett.82.21. arXiv:hep-ph/9809291.

[24] Y. Aoki, F. Csikor, Z. Fodor, A. Ukawa, The endpoint of the first order phase transition of the SU(2) gauge Higgs model on a four-dimensional isotropic lattice, Phys. Rev. D 60 (1999) 013001. http://dx.doi.org/10.1103/PhysRevD.60.013001. arXiv:hep-lat/9901021.

[25] R. Barate, et al., Search for the standard model Higgs boson at LEP, Phys. Lett. B 565 (2003) 61–75. http://dx.doi.org/10.1016/S0370-2693(03)00614-2. arXiv:hep-ex/0306033.

[26] N. Turok, J. Zadrozny, Electroweak baryogenesis in the two doublet model, Nuclear Phys. B 358 (1991) 471–493. http://dx.doi.org/10.1016/0550-3213(91)90356-3.

[27] K. Funakubo, A. Kakuto, K. Takenaga, The effective potential of electroweak theory with two massless Higgs doublets at finite temperature, Progr. Theoret. Phys. 91 (1994) 341–352. http://dx.doi.org/10.1143/PTP.91.341. arXiv:hep-ph/9310267.

[28] A. Davies, C. Froggatt, G. Jenkins, R. Moorhouse, Baryogenesis constraints on two Higgs doublet models, Phys. Lett. B 336 (1994) 464–470. http://dx.doi.org/10.1016/0370-2693(94)90559-2.

[29] J.M. Cline, K. Kainulainen, A.P. Vischer, Dynamics of two Higgs doublet CP violation and baryogenesis at the electroweak phase transition, Phys. Rev. D 54 (1996) 2451–2472. http://dx.doi.org/10.1103/PhysRevD.54.2451. arXiv:hep-ph/9506284.

[30] M. Laine, K. Rummukainen, Two Higgs doublet dynamics at the electroweak phase transition: a nonperturbative study, Nuclear Phys. B 597 (2001) 23–69. http://dx.doi.org/10.1016/S0550-3213(00)00736-7. arXiv:hep-lat/0009025.

[31] L. Fromme, S.J. Huber, M. Seniuch, Baryogenesis in the two-Higgs doublet model, J. High Energy Phys. 0611 (2006) 038. http://dx.doi.org/10.1088/1126-6708/2006/11/038. arXiv:hep-ph/0605242.

[32] J.M. Cline, K. Kainulainen, M. Trott, Electroweak baryogenesis in two Higgs doublet models and B meson anomalies, JHEP 1111 (2011) 089. arXiv:1107.3559.

[33] S.R. Coleman, The fate of the false vacuum. 1. Semiclassical theory, Phys. Rev. D 15 (1977) 2929–2936. http://dx.doi.org/10.1103/PhysRevD.15.2929, 10.1103/PhysRevD.16.1248.

[34] J. Callan, G. Curtis, S.R. Coleman, The fate of the false vacuum. 2. First quantum corrections, Phys. Rev. D 16 (1977) 1762–1768. http://dx.doi.org/10.1103/PhysRevD.16.1762.

[35] A.D. Linde, Fate of the false vacuum at finite temperature: theory and applications, Phys. Lett. B 100 (1981) 37. http://dx.doi.org/10.1016/0370-2693(81)90281-1.

[36] M. Quiros, Finite temperature field theory and phase transitions, Based on Lectures Given at the Summer School in High Energy Physics and Cosmology, ICTP, Trieste, Italy, 29 June–17 July 1998–1999, pp. 187–259. arXiv:hep-ph/9901312.

[37] G.D. Moore, K. Rummukainen, Electroweak bubble nucleation, nonperturbatively, Phys. Rev. D 63 (2001) 045002. http://dx.doi.org/10.1103/PhysRevD.63.045002. arXiv:hep-ph/0009132.

[38] G.D. Moore, K. Rummukainen, A. Tranberg, Nonperturbative computation of the bubble nucleation rate in the cubic anisotropy model, J. High Energy Phys. 0104 (2001) 017. arXiv:hep-lat/0103036.

[39] R. Apreda, M. Maggiore, A. Nicolis, A. Riotto, Gravitational waves from electroweak phase transitions, Nuclear Phys. B 631 (2002) 342–368. http://dx.doi.org/10.1016/S0550-3213(02)00264-X. arXiv:gr-qc/0107033.

[40] J.M. Cline, G.D. Moore, G. Servant, Was the electroweak phase transition preceded by a color broken phase? Phys. Rev. D 60 (1999) 105035. http://dx.doi.org/10.1103/PhysRevD.60.105035. arXiv:hep-ph/9902220.

[41] P. John, Bubble wall profiles with more than one scalar field: a numerical approach, Phys. Lett. B 452 (1999) 221–226. http://dx.doi.org/10.1016/S0370-2693(99)00272-5. arXiv:hep-ph/9810499.

[42] T. Konstandin, S.J. Huber, Numerical approach to multi dimensional phase transitions, J. Cosmol. Astropart. Phys. 0606 (2006) 021. http://dx.doi.org/10.1088/1475-7516/2006/06/021. arXiv:hep-ph/0603081.

[43] J.-H. Park, Constrained potential method for false vacuum decays, J. Cosmol. Astropart. Phys. 1102 (2011) 023. http://dx.doi.org/10.1088/1475-7516/2011/02/023. arXiv:1011.4936.

[44] S. Profumo, L. Ubaldi, C. Wainwright, Singlet scalar dark matter: monochromatic gamma rays and metastable vacua, Phys. Rev. D 82 (2010) 123514. http://dx.doi.org/10.1103/PhysRevD.82.123514. arXiv:1009.5377.

[45] C. Wainwright, S. Profumo, M.J. Ramsey-Musolf, Gravity waves from a cosmological phase transition: gauge artifacts and daisy resummations, Phys. Rev. D 84 (2011) 023521. http://dx.doi.org/10.1103/PhysRevD.84.023521. arXiv:1104.5487.

[46] J. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.

[47] S. Weinberg, New approach to the renormalization group, Phys. Rev. D 8 (1973) 3497–3509. http://dx.doi.org/10.1103/PhysRevD.8.3497.

[48] G. 't Hooft, Dimensional regularization and the renormalization group, Nuclear Phys. B 61 (1973) 455–468. http://dx.doi.org/10.1016/0550-3213(73)90376-3.

[49] D.J. Gross, R.D. Pisarski, L.G. Yaffe, QCD and instantons at finite temperature, Rev. Math. Phys. 53 (1981) 43. http://dx.doi.org/10.1103/RevModPhys.53.43.

[50] R.R. Parwani, Resummation in a hot scalar field theory, Phys. Rev. D 45 (1992) 4695. http://dx.doi.org/10.1103/PhysRevD.45.4695,10.1103/PhysRevD.48.5965. arXiv:hep-ph/9204216.