

ELENA : a software for fast and precise computation of first order phase transitions and gravitational waves production in particle physics models

Francesco Costa¹, Jaime Hoefken Zink², Michele Lucente^{3,4},
Silvia Pascoli^{3,4} and Salvador Rosauero-Alcaraz⁴

¹ *Institute of Particle and Nuclear Physics, Faculty of Mathematics and Physics, Charles University in Prague, V Holešovičkách 2, 180 00 Praha 8, Czech Republic*

² *National Centre for Nuclear Research, Pasteura 7, Warsaw, PL-02-093, Poland*

³ *Dipartimento di Fisica e Astronomia, Università di Bologna,
via Irnerio 46, 40126 Bologna, Italy*

⁴ *INFN, Sezione di Bologna, viale Berti Pichat 6/2, 40127 Bologna, Italy*

francesco.costa@matfyz.cuni.cz, jaime.hoefkenzink@ncbj.gov.pl, michele.lucente@unibo.it,
silvia.pascoli@unibo.it, rosaueroa@bo.infn.it

Abstract

We present ELENA (EvaLuator of tunnelliNg Actions), an open-source Python package designed to compute the full evolution of first-order phase transitions in the early Universe generated by particle physics models, taking into account several refinements that go beyond commonly assumed simplifications. The core of ELENA is based on a vectorized implementation of the tunnelling potential formalism, which allows for a fast computation of the finite-temperature tunnelling action. This, in turn, enables the sampling of the full range of temperatures where two phases coexist and the use of integral expressions that track the complete evolution of the transition, providing a comprehensive picture of it. In addition, ELENA provides all the tools to compute the resulting stochastic gravitational waves spectrum, allowing for the full chain of computations – from the Lagrangian parameter inputs to the final gravitational waves spectrum – in a fast and self-contained implementation.

Code availability: ELENA is available at <https://github.com/michelelucente/ELENA>.

Contents

1	Introduction	3
2	The tunnelling potential formalism	4
3	From the Lagrangian model parameters to the gravitational wave spectrum using ELENA	7
3.1	Construction of the finite temperature scalar potential	9
3.2	Computation of the critical temperatures	10
3.3	Computation of the tunnelling action	12
3.4	Computation of the fraction of Universe in the true and false vacua	14
3.5	Computation of transition milestone temperatures	19
3.6	Computation of thermal parameters determining the GW production	25
3.6.1	Transition strength α and efficiency parameters $\alpha_\infty, \alpha_{eq}$	26
3.6.2	Mean bubble separation and transition duration	31
3.7	Computation of the gravitational waves spectrum	34
4	Use case: MCMC analysis of a model using PTArcade	36
5	Performance and comparison with CosmoTransitions	38
6	Conclusion	41
A	Installation example using the conda package manager	43

1 Introduction

First-order phase transitions (FOPTs) in the early Universe provide a unique window into physics beyond the Standard Model (BSM). FOPTs are both theoretically motivated by many Standard Model (SM) extensions and observationally compelling, given that they can source a detectable stochastic gravitational wave background (SGWB). Indeed, strong FOPTs can be observed by future interferometers such as LISA, Einstein Telescope, Taiji, and Tian. Moreover, recent observation of the Hellings-Downs angular correlation pattern in multiple Pulsar Timing Array (PTA) datasets [1–4], including those from NANOGrav, EPTA, PPTA, and IPTA, has provided the first evidence for the existence of a SGWB in the nHz frequency range. Although binary systems of supermassive black holes (SMBHs) provide a plausible astrophysical interpretation of the observed signal [5–7], cosmological first-order phase transitions have emerged as one of the leading theoretical scenarios, offering a well-motivated cosmological origin for the SGWB and a better fit to the data [8–18].

A first-order phase transition arises when a scalar field acquires a non-zero vacuum expectation value. It proceeds via the nucleation of bubbles of a true vacuum within a metastable background, triggered by quantum or thermal fluctuations. The dynamics of bubble nucleation and expansion, the profile of the effective potential at finite temperature, and the tunnelling action all play crucial roles in determining the evolution of the transition and its observable consequences. While existing tools like CosmoTransitions [19], BSMPT [20], TransitionListener [21], PT2GWFinder [22], and PhaseTracer [23] can tackle some or all of these problems, they all rely on the bounce action method, which can be slow or numerically unstable for some regions of the model parameter space.

We present ELENA (EvaLuator of tunnelliNg Actions), a new open-source Python package designed to provide fast and precise computations for FOPTs induced by scalar field potentials from particle physics models. ELENA builds upon the tunnelling potential formalism [24, 25], an alternative to the traditional bounce method that recasts the vacuum decay problem as a minimization task, converting an equation of motion problem into an optimization one. This formulation provides significant numerical advantages: it is faster and more stable, and therefore is naturally suited for scans over parameter spaces, which are often required in phenomenological studies of new physics.

In addition to computing the finite-temperature tunnelling action, ELENA tracks the full evolution of the phase transition, determining key-milestone temperatures (namely critical, nucleation, percolation, and completion temperatures), and computing the thermal parameters relevant to GW generation: the strength and speed of the transition, the mean bubble separation, or the plasma sound speed, among others. The package provides a complete pipeline from Lagrangian inputs to GW spectra, making it a practical and powerful tool for model building and phenomenology.

ELENA provides ready-to-run examples and accepts potentials generated from the widely used CosmoTransitions model class, thus ensuring back-compatibility with a large number of already implemented models.

ELENA thus complements and enhances the current ensemble of computational tools in the field, contributing to the broader effort to probe early Universe physics via upcoming GW observatories.

2 The tunnelling potential formalism

The core module of **ELENA** is the implementation of the tunnelling potential formalism [24]¹, which is used to compute the tunnelling action for the decay of a scalar field ϕ from a metastable false vacuum state ϕ_+ , to a true vacuum state ϕ_- . In the seminal work [26], it was demonstrated that, under the assumption that solutions are $O(d)$ -symmetric and given a potential $V(\phi)$, the field configuration $\phi_b(\rho)$ that interpolates between ϕ_+ and the basin of ϕ_- obeys the so-called ‘bounce equation’

$$\ddot{\phi}(\rho) + \frac{d-1}{\rho} \dot{\phi}(\rho) = V'(\phi), \quad (1)$$

where the dot and prime represent the derivative with respect to ρ and ϕ , respectively, and $\rho = \sqrt{\tau^2 + |\vec{x}|^2}$ is the Euclidean radius in d dimensions (with τ the Euclidean time). Equation (1) is complemented by the boundary conditions

$$\dot{\phi}_b(0) = 0, \quad \lim_{\rho \rightarrow \infty} \phi_b(\rho) = \phi_+, \quad (2)$$

which ensures that the solution remains non-singular at the origin and that the field remains in the false vacuum state far from the nucleated bubble. Once the solution $\phi_b(\rho)$ is known, the tunnelling action can be determined by inserting it into the d -dimensional Euclidean action

$$S_{E,d} = \frac{2\pi^{d/2}}{\Gamma(d/2)} \int_0^\infty \left[\frac{1}{2} \dot{\phi}_b^2 + V(\phi_b) - V(\phi_+) \right] \rho^{d-1} d\rho. \quad (3)$$

Equation (1) with the boundary conditions in Equation (2) has an intuitive classical mechanics interpretation: by identifying ϕ as the position of a classical particle subject to a potential $-V$ and to a friction term that decreases over time as ρ^{-1} , with ρ being the time variable, the solution to the problem is equivalent to finding the position ϕ_0 such that the particle, starting at rest from $\phi = \phi_0$ at $\rho = 0$, rolls down and exactly stops at $\phi = \phi_+$. It is clear from Figure 1 that the solution to this problem is an unstable point [26]: values of ϕ slightly larger than ϕ_0 will result in the particle overshooting the point ϕ_+ and rolling away from it, while slightly smaller values make the particle roll back and oscillate around the local minimum of $-V$ (corresponding to the maximum of V). This observation is the basis for a popular numerical recipe [19] to solve Equations (1) and (2), known as the overshoot-undershoot method: it amounts to solving the equations of motion for a trial value of ϕ_0 . Depending on the asymptotic behaviour of the field value (that is, depending on whether the trajectory undershoots or overshoots ϕ_+), the initial position ϕ_0 is displaced closer or further from the top. This procedure is iterated until the required numerical precision is reached.

The undershoot-overshoot method can be numerically demanding, given that small deviations from the correct value of ϕ_0 result in large differences of the asymptotic field value from ϕ_+ . In addition, the convergence time of the solution tends to infinity for infinitesimal energy differences between the false and true vacua, although analytical solutions for this regime are known [26].

¹This formalism has also been extended to multi-field potentials [25], although **ELENA** focuses on one-field potentials in its first version.

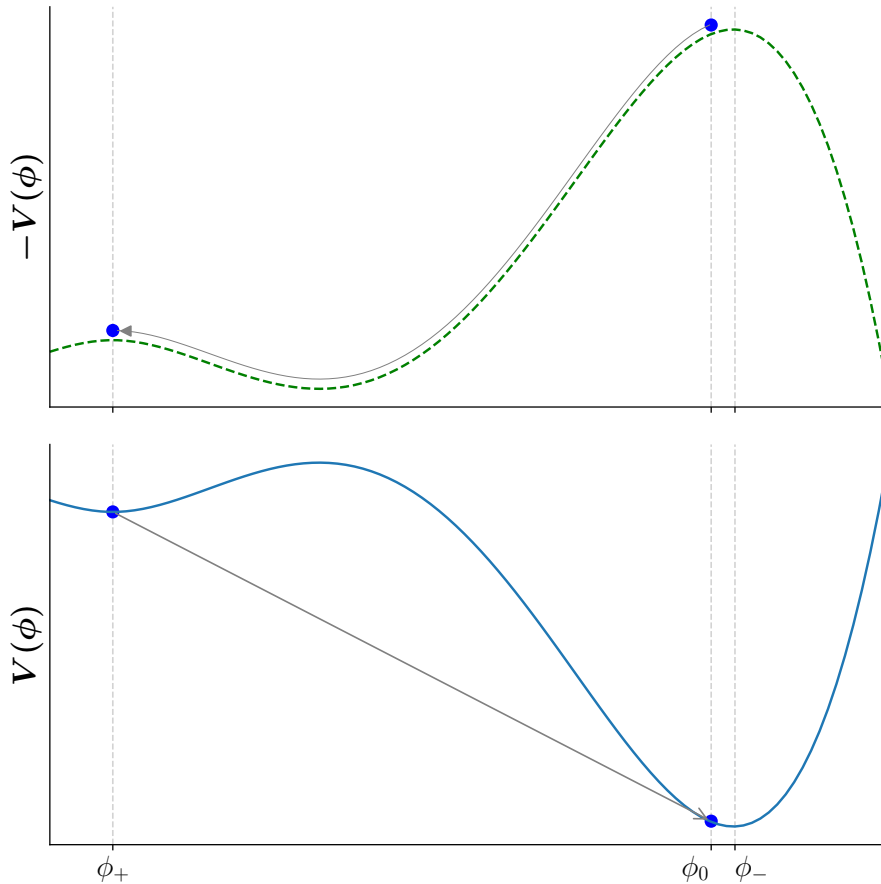


Figure 1: Pictorial representation of the classical mechanical dual of the bounce equations of motion, Equations (1) and (2). *Upper panel:* a classical particle at position ϕ is subject to the potential $-V(\phi)$, with a friction term that decreases with time ρ (see Equation (1)). The solution satisfying the boundary conditions in Equation (2) amounts to finding the value of ϕ_0 such that the particle, starting at rest from $\phi = \phi_0$ at $\rho = 0$, stops at $\phi = \phi_+$ for $\rho \rightarrow \infty$. *Lower panel:* the scalar field nucleates via an $O(d)$ -symmetric field configuration, with value ϕ_0 at $\rho = 0$ and ϕ_+ at $\rho \rightarrow \infty$, where ρ is the Euclidean radius in d dimensions. The field profile at finite values of ρ is determined by the solution $\phi_b(\rho)$ to Equations (1) and (2).

The tunnelling potential formalism [24] addresses the numerical issues of the bounce formalism by redefining the equations of motion, such that the correct value of ϕ_0 becomes a minimum of the so-called tunnelling action. The numerical advantage is twofold: first and foremost, finding the minimum of a function is much easier than finding a saddle point. Moreover, it is possible to completely remove the dependency on the bounce equations Equations (1) and (2), so that no differential equation needs to be numerically integrated.

The tunnelling potential is an auxiliary function defined as

$$V_t(\phi) \equiv V(\phi) - \frac{1}{2}\dot{\phi}_b^2, \quad (4)$$

where ϕ_b is the solution to the bounce equations (Equations (1) and (2)). As demonstrated in Ref. [24], $V_t(\phi)$ is a monotonic function with $V_t(\phi) \leq V(\phi)$, and is defined only in the range $\phi \in [\phi_+, \phi_0]$, with $V_t = V$ at the endpoints. As anticipated, it is possible to remove the dependence on the bounce

solution ϕ_b by employing the relation

$$\dot{\phi}_b = -\sqrt{2[V(\phi) - V_t(\phi)]}, \quad (5)$$

as well as on the radial variable by using

$$\rho = (d-1) \sqrt{\frac{V - V_t}{(V'_t)^2}}. \quad (6)$$

The new equations of motion are given by

$$(V'_t)^2 = \frac{d-1}{d} [V'V'_t - 2(V_t - V)V''_t], \quad (7)$$

with boundary conditions

$$V_t(\phi_+) = V(\phi_+), \quad V_t(\phi_0) = V(\phi_0). \quad (8)$$

The new problem is to find the correct value of ϕ_0 and the form of $V_t(\phi)$ that solves Equations (7) and (8). Although this does not seem an easier task with respect to the starting point in Equations (1) and (2), as demonstrated in Ref. [24], one does not need to solve this differential equation in practice. An excellent approximation for V_t can indeed be derived by considering the shape of the scalar potential V ; together with the observation that ϕ_0 is a minimum of the tunnelling action $S_{E,d}$, given by

$$S_{E,d} = \frac{(d-1)^{(d-1)} (2\pi)^{\frac{d}{2}}}{\Gamma\left(1 + \frac{d}{2}\right)} \int_{\phi_+}^{\phi_0} \frac{(V - V_t)^{\frac{d}{2}}}{|V'_t|^{(d-1)}} d\phi, \quad (9)$$

this allows to apply the following numerical algorithm to solve the problem:

Construct an approximation to $V_t(\phi)$ following the procedure outlined in Ref. [24], and then insert it in Equation (9), computing $S_{E,d}$ as a function of ϕ_0 . The solution to the problem is given by the value of ϕ_0 that minimises $S_{E,d}$.

This is exactly the numerical procedure implemented in **ELENA**, contained in the class `espinosa.Vt_vec` and described in more detail in the following Sections.

In Figure 2 we report an example of the formalism. In the left panel, we show the potential $V(\phi)$ (black line) together with the tunnelling potential $V_t(\phi, \tilde{\phi}_0)$. This notation means that V_t is constructed following the prescription in Ref. [24] for an arbitrary value $\tilde{\phi}_0$ of the field at the centre of the nucleated bubble, which is not necessarily the solution to the equations of motion. The correct form for V_t , which is the one constructed by choosing the value $\tilde{\phi}_0 = \phi_0$ minimising the tunnelling action, is shown as a blue line. We notice that, for values of $\tilde{\phi}_0$ that are excessively far-away from the correct one, the construction in Ref. [24] breaks down as V_t is no longer monotonically decreasing; we identify these problematic tunnelling potentials by gray lines, while the well-behaved ones are identified by orange lines. This behaviour is nevertheless not an issue for the numerical routine, since the action $S_{E,d}(\tilde{\phi}_0)$

sharply increases when moving towards this problematic region (see right panel in Figure 2) such that the results of the calculation within the region itself systematically give values of the tunnelling action that are orders of magnitude larger than the minimum of $S_{E,d}$ at ϕ_0 (even if the result of the computation cannot be trusted). We conclude that, since ELENA looks for the minimum of $S_{E,d}$ as a function of $\tilde{\phi}_0$, the results are reliable: the algorithm works well in the region of interest, while in regions where the method is unreliable, the computed value of $S_{E,d}$ remains substantially larger than its minimum, preventing these configurations from being misidentified as valid solutions to the equations of motion.

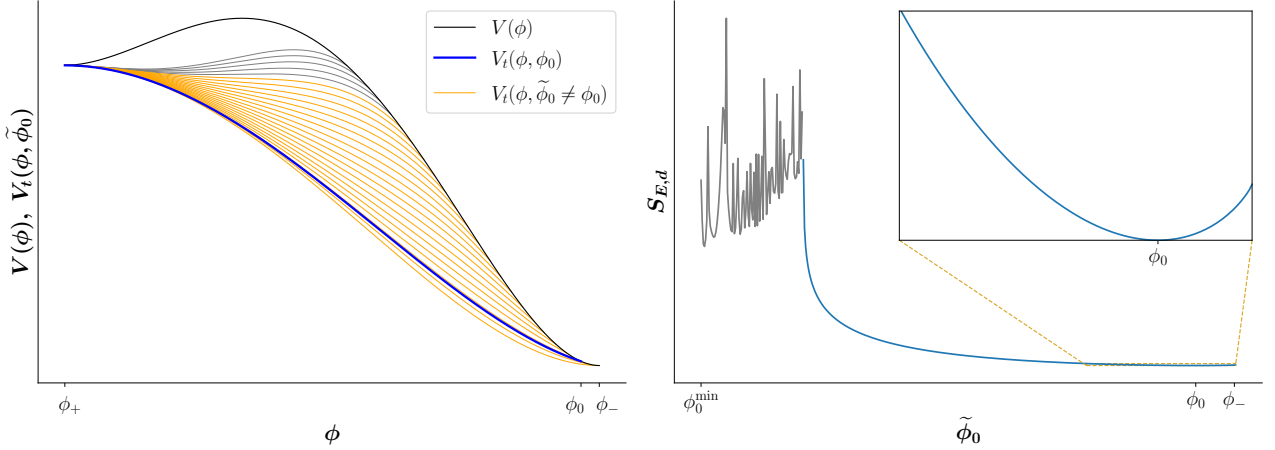


Figure 2: Example of the tunnelling potential formalism. *Left panel:* the tunnelling potential $V_t(\phi, \tilde{\phi}_0)$ for different values of the $\tilde{\phi}_0$ parameter, obtained by following the procedure outlined in Ref. [24]; the blue line indicates the potential $V_t(\phi, \phi_0)$, obtained by choosing the value of $\tilde{\phi}_0 = \phi_0$ that minimises the action $S_{E,d}(\tilde{\phi}_0)$. Gray lines represent configurations where the construction of the tunnelling potential breaks down, while orange lines are numerically viable configurations. For reference, the potential $V(\phi)$ is plotted as a black line. *Right panel:* the value of the action $S_{E,d}(\tilde{\phi}_0)$ as a function of the $\tilde{\phi}_0$ parameter, computed following Equation (9). $\phi_0^{\min}$ is the smallest value of $\tilde{\phi}_0$ for which the tunnelling can take place, that is, the smallest field value in the region to the right of the local barrier for which $V(\phi) < V(\phi_+)$. Values in gray correspond to unreliable V_t constructions (cf. the gray lines in the left panel), while the blue line corresponds to numerically reliable computations. The inset box magnifies the curve around $\tilde{\phi}_0 = \phi_0$, the value of $\tilde{\phi}_0$ that minimises $S_{E,d}$.

3 From the Lagrangian model parameters to the gravitational wave spectrum using ELENA

In this Section we outline how to use ELENA to perform the full chain of computations necessary to predict the SGWB from a first order phase transition, inputting only the model Lagrangian parameters. In each step, we describe the relevant modules used by ELENA. The full code is available as a ready-to-run Jupyter notebook in `./examples/phase_transition.ipynb`. We summarise in Table 1 ELENA's main features, and compare with those from other popular codes [19, 20, 22, 23, 27–29].

Software	Phases	S_3/T & Method	T_n	T_p	T_e	GW	Assumptions
ELENA	✓	✓ Single field Tunneling potential	✓	✓	✓	✓	Adiabatic expansion
CosmoTransitions [19]	✓	✓ Multi-field Shooting and path deformation	✓	✗	✗	✗	$S_{E,3}(T_n)/T_n = 140$
TransitionListener [21]	✓	✓ Multi-field Shooting and path deformation	✓	✗	✗	✓	Adiabatic expansion $H(T)^2 \propto \rho_{\text{rad}}(T)$ $\Gamma(T_n)/H^4(T_n) = 1$ $\Gamma(t) \sim \Gamma_n e^{\beta t}$ GW at T_n
BubbleProfiler [27]	✗	✓ Multi-field Shooting and perturbative	✗	✗	✗	✗	None
PhaseTracer2 [23]	✓	✓ Multi-field Shooting and path deformation	✓	✗	✗	✓	Adiabatic expansion $H(T)^2 \propto \rho_{\text{rad}}(T)$ $S_{E,3}(T_n)/T_n = 140$ $\Gamma(t) \sim \Gamma_n e^{\beta t}$ GW at T_n
BSMPT [20]	✓	✓ Multi-field Shooting and path deformation	✓	✓	✓	✓	Adiabatic expansion $H(T)^2 \propto \rho_{\text{rad}}(T)$ Constant g_* MIT bag model $\Gamma(t) \sim \Gamma_n e^{\beta t}$ GW from sound waves & turbulence
FindBounce [29]	✗	✓ Multi-field Polygonal bounces	✗	✗	✗	✗	None
PT2GWFinder [22]	✓	✓ Single field Polygonal bounces	✓	✓	✓	✓	Adiabatic expansion Constant g_* MIT bag model $\Gamma(t) \sim \Gamma_n e^{\beta t}$

Table 1: Comparison of the features included in ELENA and other popular codes, along with the approximations adopted in their pipelines. We use different colors for the name of each software to highlight the language in which they are coded. We use blue for Python, orange for C++, and violet for Mathematica. We note that both PT2GWFinder and PhaseTracer2 can be used together with DRAIgo [30]. Moreover, while PhaseTracer2 uses the nucleation temperature to compute the GW spectrum, we point out the existence of TransitionSolver, developed by the same authors, which can, in principle, compute all parameters. Given that the latter is still in developing stages, we refrain from comparing to it. TransitionListener uses CosmoTransitions to find the phases and to compute the action, adding GW-related features to it.

3.1 Construction of the finite temperature scalar potential

ELENA uses the `model` class from CosmoTransitions [19] to construct the finite-temperature scalar potential. We thus refer the user to the CosmoTransitions documentation for details on the implementation of a new model. Once the potential has been constructed, ELENA uses its own modules for the computation of the phase transition dynamics, not relying on CosmoTransitions any longer.

The current version of ELENA includes the implementation of the dark sector model presented in Ref. [31], composed of a complex scalar ϕ , charged under a new $U(1)_D$ gauge symmetry, and its associated dark photon Z'_μ . Its Lagrangian is given by

$$\mathcal{L} = (D_\mu \phi)^* (D^\mu \phi) - \frac{1}{4} Z'_{\mu\nu} Z'^{\mu\nu} - V(\phi^* \phi), \quad V = -\mu_\phi^2 \phi^* \phi + \lambda_\phi (\phi^* \phi)^2, \quad (10)$$

where $D_\mu \equiv \partial_\mu - i\sqrt{2}g_D Z'_\mu$ is the covariant derivative and g_D the dark gauge coupling. The potential in Equation (10) with $\mu_\phi^2, \lambda_\phi > 0$ has a global minimum at $v_\phi^0 = \mu_\phi / \sqrt{\lambda_\phi}$; when the scalar field acquires a vacuum expectation value (vev) v_ϕ^0 , the $U(1)_D$ symmetry is spontaneously broken, resulting in the mass spectrum $m_{Z'}^2 = g_D^2 (v_\phi^0)^2$ and $m_\phi^2 = 2\lambda_\phi (v_\phi^0)^2$.

The implementation of the potential in Equation (10) provided in ELENA employs the on-shell renormalisation method, which is expected to give more conservative predictions for the resulting SGWB [32]. Consequently, the quantities appearing in the Lagrangian match the observable physical ones.

Once a model class has been implemented, the finite temperature potential is created by simply initialising an instance of the class with the desired Lagrangian parameters, as shown in the Listing 1.

```

1 from model import model
2
3 # Select the value of the model parameters
4 lambda_ = 1.65e-3
5 g = 0.54
6 vev = 500
7
8 # This constructs the finite-temperature potential as an instance of the class model
9 dp = model(vev, lambda_, g, xstep = 1e-3 * vev, Tstep = 1e-3 * vev) # Dark photon
   model instance
10 V = dp.DVtot # Scalar potential
11 dV = dp.gradV # Gradient of the scalar potential

```

Listing 1: Finite temperature potential implementation and initialization in ELENA.

The arguments `xstep` and `Tstep` are internal arguments of CosmoTransitions. We find it useful to normalise them to the energy scale of the problem (i.e. the vev value) to improve the numerical stability of the computation.

In the following, we present two examples of a phase transition computation for different choices of the model parameters: the first one is dubbed “Fast”, given that the transition takes place on a timescale much smaller than the Hubble time; in the second one, dubbed “Slow”, the duration of the transition is instead comparable to it (cf. Section 3.6.2 for numerical values). The input Lagrangian parameters

for each case are collected in Table 2.

Parameter	Fast	Slow
λ_ϕ	$1.65 \cdot 10^{-3}$	$6 \cdot 10^{-3}$
g_D	0.54	0.75002
v_ϕ^0	500 MeV	500 MeV

Table 2: Input parameters for the two example benchmark points analysed in this work, as referred to the Lagrangian from Equation (10). The “Slow” point coincides with the BP1 in Ref. [31].

3.2 Computation of the critical temperatures

A FOPT can only happen if a barrier separating the true and false vacua is present. This condition determines the range of temperatures $T \in [T_{min}, T_c]$ for which a FOPT is possible: T_c is the critical temperature, defined as the temperature at which the values of the potential in the true and false vacua are degenerate and a potential barrier separates the two minima; T_{min} is the temperature below which a potential barrier is no longer present. For $T > T_c$, the global minimum of the potential is the one corresponding to the false vacuum; for $T < T_{min}$ the field can simply roll to the true minimum, a process that does not result in bubble nucleation.

ELENA implements its own routines, `temperatures.find_T_min` and `temperatures.find_T_max`, to compute the temperature range in which the FOPT is possible. The algorithm looks for changes in the sign of the gradient of the potential to find the local extrema of the potential itself. Next, it checks if a FOPT is possible given the shape of the potential at a given temperature. The routines `temperatures.find_T_min` and `temperatures.find_T_max` loop over different temperature values (starting from small and large temperatures, respectively) to verify the smallest and largest temperatures for which a FOPT can take place, reducing the size of the step at each iteration until a desired precision is met. Once a first estimation from `temperatures.find_T_min` is known, ELENA provides the function `temperatures.refine_Tmin` to further refine the determination of T_{min} by looping over a narrower range of temperatures using smaller steps. This allows to obtain an accurate result without a significant numerical overload. Such a precise determination can be necessary for very long transitions, when the completion temperature is close to T_{min} . We do not implement an analogous function to refine T_c because the nucleation probability is infinitesimal at T_c , and we find that the computation of the transition dynamics does not change appreciably with a more precise determination of T_c with respect to the one already obtained by `temperatures.find_T_max`. An example computation of the critical temperatures using ELENA is shown in the Listing 2.

```

1 from temperatures import find_T_min, find_T_max, refine_Tmin
2
3 T_max, vevs_max, max_min_vals, false_min_tmax = find_T_max(V, dV, precision = 1e-2,
4   Phimax = 2 * vev, step_phi = vev * 1e-2, tmax = 2.5 * vev)
5
6 T_min, vevs_min, false_min_tmin = find_T_min(V, dV, tmax = T_max, precision = 1e-2,
7   Phimax = 2 * vev, step_phi = vev * 1e-2, max_min_vals = max_min_vals)
8
9 if T_max is not None and T_min is not None:

```

```

7     maxvev = np.max(np.concatenate((vevs_max, vevs_min)))
8 elif T_max is not None:
9     maxvev = np.max(vevs_max)
10 elif T_min is not None:
11     maxvev = np.max(vevs_min)
12 else:
13     maxvev = None
14
15 T_min = refine_Tmin(T_min, V, dV, maxvev, log_10_precision = 6) if T_min is not None
    else None

```

Listing 2: Computation of the critical and minimum temperatures.

Apart from the determination of T_c (`T_max`), the function `temperatures.find_T_max` also returns other quantities that are useful to improve the convergence of subsequent computations: a list of vev values at finite temperature (`vevs_max`), an array containing the field values where the potential $V(\phi, T_c)$ is at its global maximum and minimum (`max_min_vals`, corresponding to the locations of the barrier maximum and true minimum at T_c), and an array containing the field location and potential value at the false minimum at T_c (`false_min_tmax`). Analogously, `temperatures.find_T_min` returns the value of T_{min} and two auxiliary quantities, here saved as `vevs_min` and `false_min_tmin`. Optional arguments to the above described functions are: `precision` (default = $1e-2$) that sets the desired precision of the computation², `Phimax` (set by default to 150) that sets the maximum value of the field to be considered, `step_phi` (default 0.1) that sets the step size in the field dimension when constructing the potential and gradient, and `tmax` and `tmin` (default 250 and 0), which set the maximal and minimal temperatures to be considered in the search.³ The optional argument `max_min_vals`, only present in `temperatures.find_T_min`, can improve the computation by providing information on the barrier and true minimum locations.

Once the value of T_{min} has been determined by `temperatures.find_T_min`, it can be refined by using `temperatures.refine_Tmin`, which checks the minimum value of the form `T_min * (1 - k * 10**(-log_10_precision))`, with `k` being an integer, for which a barrier is still present. In contrast to the routine in `temperatures.find_T_min`, the function `temperatures.refine_Tmin` only uses the size of the temperature step as a criterion to stop the refinement, disregarding the height of the barrier.

The finite temperature potential for the points in Table 2 is plotted in Figure 3 for different values of the temperature, including T_c (orange dashed line) and T_{min} (red dot-dashed line), together with values at $1.1*T_c$, $(T_c - T_{min})/2$ and $T = 0$ for reference (blue, green, and purple lines, respectively). As can be seen, both the height of the barrier and the depth of the true minimum are larger in the “Slow” case, resulting in a stronger transition in this scenario. Moreover, in this case the barrier persists down to zero temperature as a result of the Coleman-Weinberg corrections to the zero temperature potential. The slow nature of the transition is typical of conformal-like potentials (cf. e.g. [33, 34]).

²The loops stop when the ratio between the height of the potential barrier and the depth of the true minimum is smaller than `precision` for `temperatures.find_T_min`, or when the difference between the potential value at true and false minima divided by the height of the barrier is smaller than `precision` for `temperatures.find_T_max`.

³Given the scale invariance of the potential, we do not specify the units on these dimensionful variables, but note that these are assumed to be the same between field and temperature variables and set by `units`.

We do not indicate the energy scale in Figure 3 because the potential enjoys the scaling symmetry

$$V(\phi, \lambda, g, v, T) = \frac{V(\xi\phi, \lambda, g, \xi v, \xi T)}{\xi^4}, \quad (11)$$

for arbitrary rescaling ξ of the dimensionful quantities, meaning that up to a global factor in V the plotted shapes are valid for arbitrary choices of the energy scale.

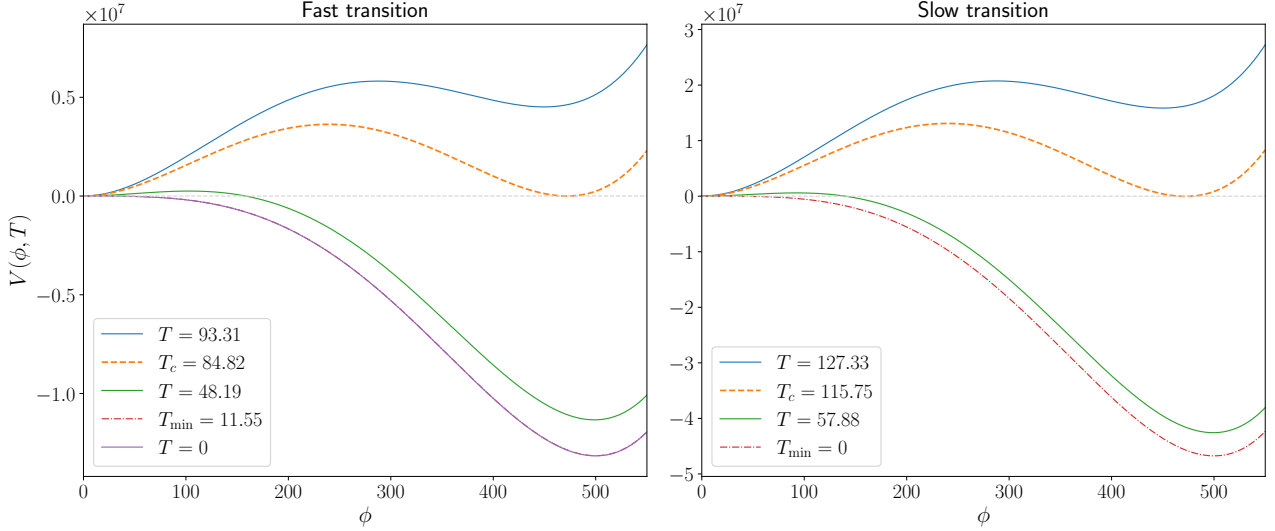


Figure 3: Shape of the finite temperature potential for the example points in Table 2 at different temperatures, corresponding to $T = 1.1 \cdot T_c$ (blue line), where T_c is the critical temperature, T_c (orange dashed-line), $T = (T_c - T_{min})/2$ (green line), where T_{min} is the smallest temperature at which a barrier is present, and T_{min} (red dot-dashed line). Additionally, when $T_{min} \neq 0$, we show the potential at $T = 0$ as a purple line. Up to a global rescaling of V , these results are valid for an arbitrary energy scale (cf. Equation (11)). The values of the temperatures shown in the legend correspond to the same ones chosen for ϕ and given in [units](#).

3.3 Computation of the tunnelling action

Once the range of temperatures $T \in [T_{min}, T_c]$ over which a FOPT is possible has been determined, the most important quantity to be computed is the tunnelling action in Equation (9), which governs the evolution of the transition. At finite temperature, it enters in the false vacuum decay rate as [26, 35, 36]

$$\Gamma(T) \simeq T^4 \left(\frac{S_{E,3}}{2\pi T} \right)^{3/2} e^{-S_{E,3}/T}. \quad (12)$$

The computation of $S_{E,3}$ in ELENA is performed using the class `espinosa.Vt_vec`, where the name `Vt_vec` stands for a vectorised implementation of the tunnelling potential algorithm discussed in Section 2. This class computes, for a given potential, the relevant quantities appearing in Equation (9) using a discrete grid of values for the ϕ and $\tilde{\phi}_0$ variables, and selects as solution the value of $\tilde{\phi}_0 = \phi_0$ that minimises $S_{E,3}$. The step size `step_phi` over which to scan ϕ is chosen by the user, while the step size over $\tilde{\phi}_0$ starts with a relatively large value exploring the full range of possible solutions; at subsequent iterations, the target space of $\tilde{\phi}_0$ is reduced around an interval containing ϕ_0 while the step size is

decreased correspondingly. The iterations continue until the relative difference in the value of the action computed in two subsequent iterations is smaller than a user-defined value given by `precision`. The discretised computation performed by `espinosa.Vt_vec` employs fully vectorised NumPy arrays [37], allowing for a numerically efficient computation. Notice that no numerical differential equation needs to be solved to determine ϕ_0 , thus allowing for a fast computation.

The computation of the tunnelling action in ELENA is done by creating an instance of the class `espinosa.Vt_vec` with the desired model parameters, and then extracting the attribute `action_over_T` from that instance. We provide an example code in the Listing 3, while stressing that `espinosa.Vt_vec` can be implemented in more general coding structures according to the specific needs of the user. In addition to `action_over_T`, an instance of `espinosa.Vt_vec` contains several other attributes that provide further quantities relevant to the tunnelling process. These are exemplified as well in the Listing 3.

```

1 import numpy as np
2 from espinosa import Vt_vec
3
4 true_vev = {}
5 S3overT = {}
6 V_min_value = {}
7 phi0_min = {}
8 V_exit = {}
9 false_vev = {}
10
11 def action_over_T(T, c_step_phi = 1e-3, precision = 1e-3):
12     instance = Vt_vec(T, V, dV, step_phi = c_step_phi, precision = precision, vev0 =
13         maxvev, int_threshold = 2e-1)
14     if instance.barrier:
15         true_vev[T] = instance.true_min
16         false_vev[T] = instance.phi_original_false_vev
17         S3overT[T] = instance.action_over_T
18         V_min_value[T] = instance.min_V
19         phi0_min[T] = instance.phi0_min
20         V_exit[T] = instance.V_exit
21         return instance.action_over_T
22     else:
23         return None
24
25 n_points = 100
26 temperatures = np.linspace(T_min, T_max, n_points)
27 action_vec = np.vectorize(action_over_T)
28 action_vec(temperatures)

```

Listing 3: Example of implementation of a tunnelling action computation.

Several Python dictionaries are created in the Listing 3 to store the result of the computation for different values of the temperature parameter. A function `action_over_T()` is then defined as a wrapper of `espinosa.Vt_vec`, to compute an instance of the class for a specific temperature and store the computed attributes in the dictionaries. Useful attributes of an instance of the class include:

- `action`: the value of the finite-temperature d -dimensional tunnelling action $S_{E,d}$;
- `action_over_T`: the value of the action divided by the temperature, i.e. $S_{E,d}/T$;
- `phi0_min`: the field value ϕ_0 minimising $S_{E,d}$ in Equation (9);
- `V_exit`: the value of the potential at $V(\phi_0)$;
- `true_min`: the value of the field ϕ at the true vacuum;
- `phi_original_false_vev`: the value of the field ϕ at the false vacuum;
- `min_V`: the value of the potential at the true vacuum.

The boolean attribute `barrier` indicates whether a barrier is present at the specified temperature.

The class `espinosa.Vt_vec` accepts the optional arguments `d` (set to 3 by default) for the number of dimensions, `vev0` (default 100, assumed to be in the same energy units as the field entering in the potential) for an initial estimate of the finite temperature vev value, `step_phi` (default `1e-3`) that sets the step size in the ϕ dimension in units of `vev0`, `precision` (default `1e-3`) determining the desired relative precision in the computation of $S_{E,d}$, `ratio_vev_step0` (default 50) that sets the initial step size in the $\tilde{\phi}_0$ dimension as `vev0 / ratio_vev_step0`. In addition, the optional argument `save_all` (default `False`) determines whether the internal computations for each value of $\tilde{\phi}_0$ should be saved as an attribute of the class instance; this can be useful if the user wants to perform consistency checks on the results or post-process these data.

After having defined the function `action_over_T()`, the rest of the code in Listing 3 simply computes the tunnelling quantities over 100 values of temperature between T_{min} and T_c . This computation, performed on an Apple M2 processor, employs approximately 2 seconds, thus amounting to approximately 20 milliseconds for each tunnelling computation. The results of the computation are reported in Figure 4 for the “Fast” transition benchmark point, and in Figure 5 for the “Slow” one. Each plot shows, as a function of the temperature T , the action $S_{E,3}/T$ (upper-left), the solution $\phi_0(T)$ (upper-right)⁴ and the potential value at this point $V(\phi_0(T), T)$ (middle-right), the position of the true vacuum $\phi_-(T)$ (bottom-left) and the minimum of the potential $V(\phi_-(T), T)$ (middle-left panel), as well as the false vacuum value $\phi_+(T)$ (bottom-right panel). The most striking difference between the two points is in the shape of the action. It monotonically decreases with temperature for the fast transition, while in the slow scenario, it shows a global minimum at $T = 11.7$ MeV.

3.4 Computation of the fraction of Universe in the true and false vacua

The key quantity that tracks the evolution of the transition at each temperature value is the fraction of the Universe in the false vacuum $P_f(T)$ (or equivalently the fraction of volume in true vacuum $P_t(T) = 1 - P_f(T)$). Bubble nucleation is a stochastic process that happens continuously as soon as it is energetically allowed, with a nucleation probability per unit volume and unit time given by Equation (12). Although for fast transitions there is usually a characteristic temperature that fully

⁴We remind the reader that ϕ_0 is the value of the field at the centre of the bubble right after it is nucleated.

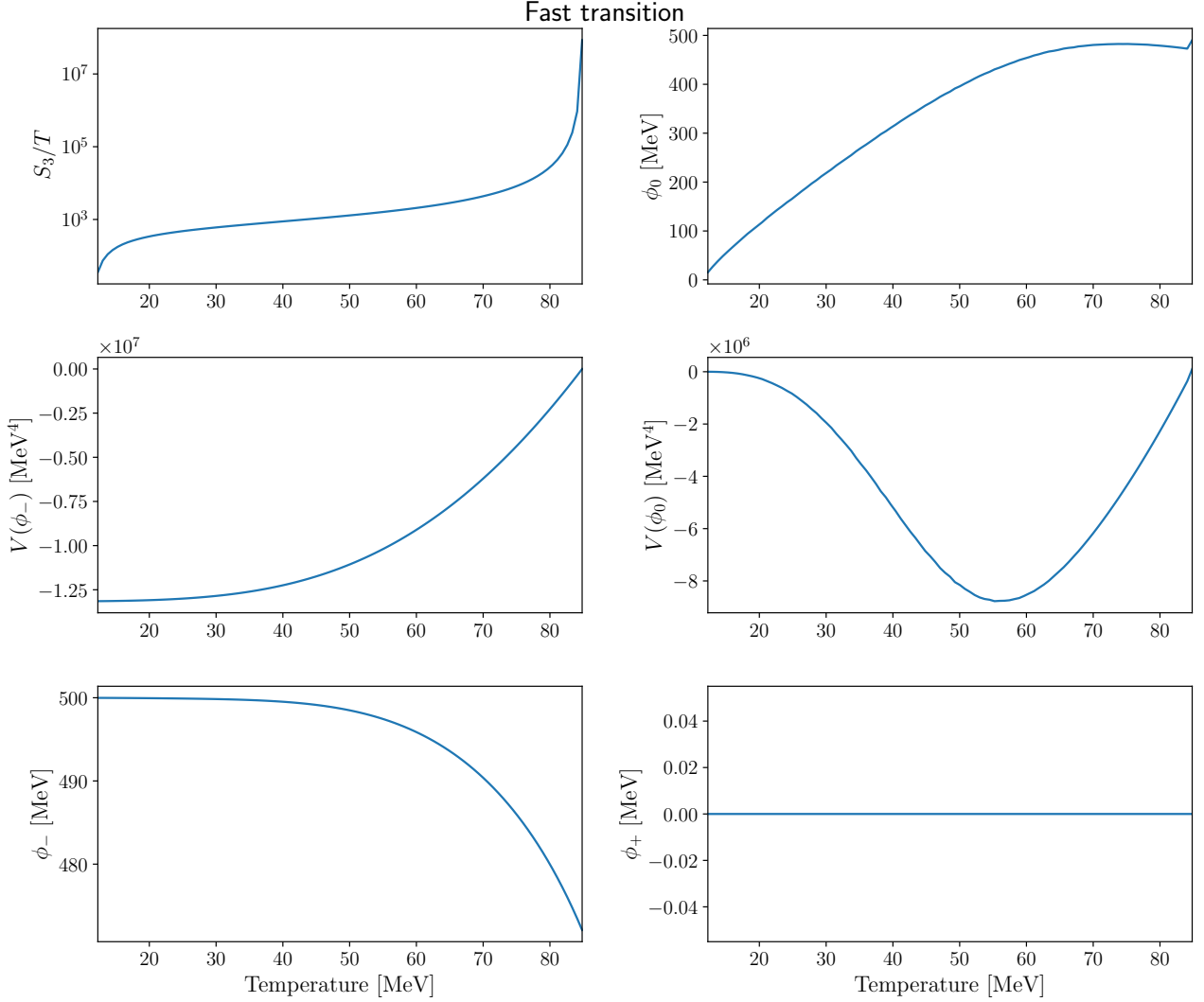


Figure 4: Evolution of the relevant quantities related to tunnelling as a function of temperature for the “Fast” point in Table 2. *Upper-left*: action $S_{E,3}/T$. *Upper-right*: tunnelling solution ϕ_0 . *Centre-left*: potential at true vacuum $V(\phi_-)$. *Centre-right*: potential at ϕ_0 , $V(\phi_0)$. *Bottom-left*: true vacuum ϕ_- . *Bottom-right*: false vacuum ϕ_+ .

defines the dynamics of the process (the so-called nucleation temperature, where most of the bubble nucleation happens), this is generally not true for slow transitions, which are phenomenologically more interesting since they produce GW signals with larger amplitudes and therefore better detection prospects.

To obtain general, reliable results, one shall thus track the evolution of $P_f(T)$ over the full range of temperatures where bubble nucleation is possible, given by $T \in [T_{min}, T_c]$. The efficient computation of the bounce action in ELENA makes it possible to study the phase transition following the evolution of $P_f(T)$, instead of relying on approximations such as comparing the nucleation rate in Equation (12) to the Hubble expansion rate, $H(T)$.

ELENA provides modules to readily compute $P_f(T)$ from the tunnelling action, without making any strong assumption on the underlying cosmology. The only underlying assumptions are the ones leading

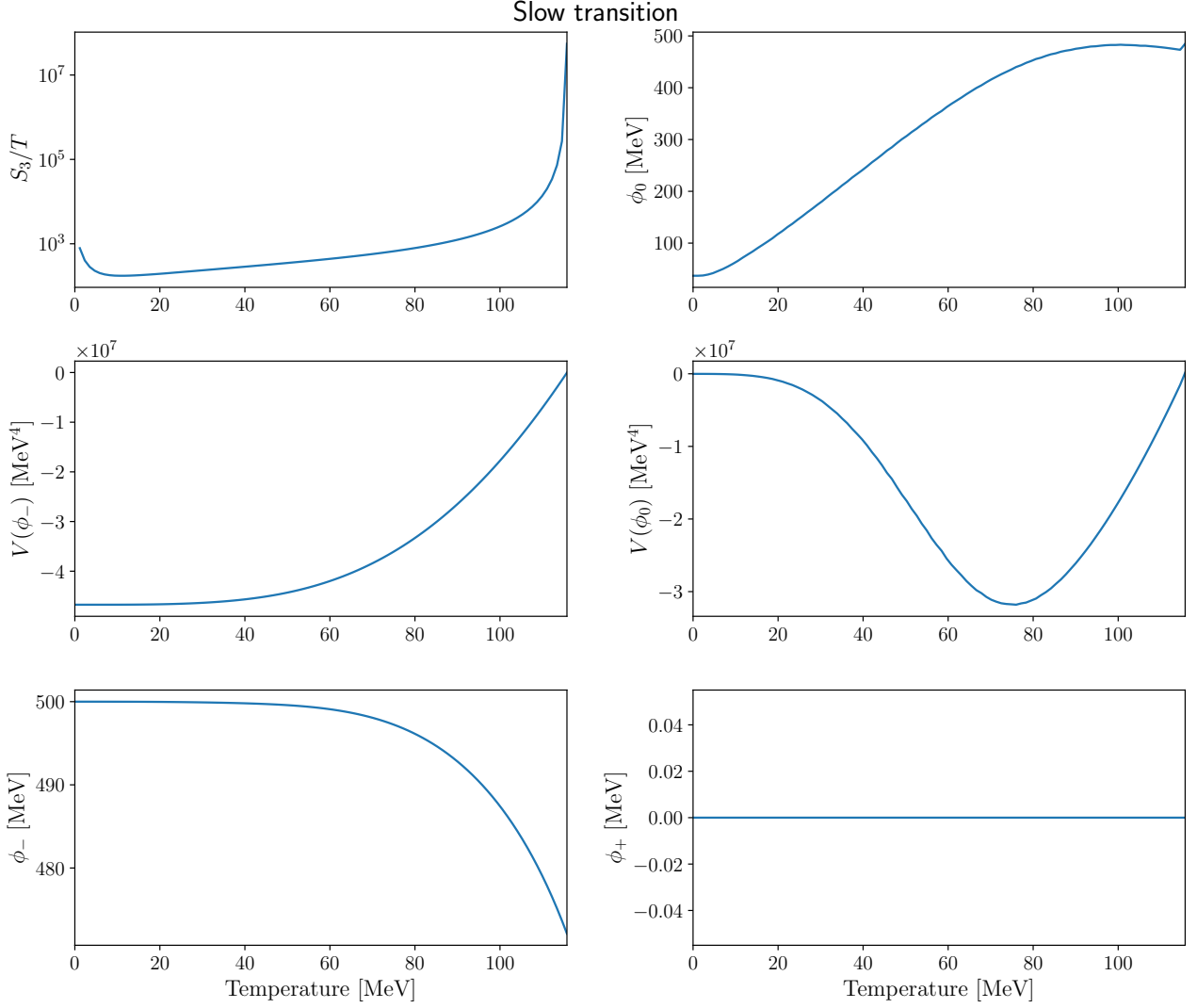


Figure 5: Same as in Figure 4, but for the “Slow” point in Table 2.

to the JMAK equation [38–44], which is given by:

$$P_f(T) = \exp\left(-\mathcal{V}_t^{ext}(T)\right), \quad (13)$$

where $\mathcal{V}_t^{ext}(T)$ is the *extended* volume in true vacuum at a given temperature. The extended volume is defined as the volume that all nucleated bubbles would occupy, assuming a nucleation probability given by $\Gamma(T)$, while neglecting the fact that a fraction of this volume may have already been converted to true vacuum at that temperature. This amounts to considering processes such as the nucleation of a new bubble inside a region of true vacuum, or double-counting the volume when two bubbles grow into each other. $\mathcal{V}_t^{ext}(T)$ clearly gives a strong overestimate of the real fraction of volume in true vacuum (from which the “extended” adjective). Nevertheless, it has been demonstrated that, under reasonable assumptions, the fraction of the Universe in the false vacuum state is correctly related to $\mathcal{V}_t^{ext}$ via Equation (13). We refer the reader to Ref. [45] for a thorough discussion on the JMAK equation in the context of cosmological phase transitions, including multiple derivations of it and a

critical analysis of the validity of its assumptions.

The extended volume in true vacuum is obtained by summing the volumes of the single bubbles nucleated up to a given time t . By assuming a homogeneous rate of bubble nucleation per unit volume and unit time $\Gamma(t)$, the corresponding expression in a static Universe reads

$$\mathcal{V}_{t,\text{static}}^{\text{ext}} = \int_{t_0}^t dt' \Gamma(t') V(t', t), \quad (14)$$

where t_0 is the time corresponding to the critical temperature T_c and $V(t', t)$ is the volume at time t of a bubble that nucleated at time t' . Under the assumption of spherically symmetric bubbles, it is given by

$$V(t, t') = \frac{4\pi}{3} R(t, t')^3, \quad (15)$$

with the radius at time t given by

$$R(t, t') = R_0(t') + \int_{t'}^t dt'' v_w(t''), \quad (16)$$

where R_0 is the initial bubble radius and v_w the bubble wall velocity. Assuming that a bubble rapidly grows to a radius much larger than $R_0(t')$, and that the wall rapidly accelerates to its final constant velocity v_w , taken to be a homogeneous parameter in the Universe, one gets

$$V(t, t') = \frac{4\pi}{3} \left[v_w \int_{t'}^t dt'' \right]^3, \quad (17)$$

where we keep the explicit form of the time integral for subsequent generalisation to the case of an expanding Universe. The expression for the extended volume thus reads

$$\mathcal{V}_{t,\text{static}}^{\text{ext}}(t) = \frac{4\pi}{3} v_w^3 \int_{t_0}^t dt' \Gamma(t') \left[\int_{t'}^t dt'' \right]^3. \quad (18)$$

The generalisation to an expanding universe is readily obtained. First, we must appropriately scale the nucleation rate to reflect the change in unit volume over time

$$\int_{t_0}^t dt' \Gamma(t') \rightarrow \int_{t_0}^t dt' \Gamma(t') \left(\frac{a(t')}{a(t)} \right)^3, \quad (19)$$

where we have chosen to normalise the unit volume at time t . Then, we must account for the fact that the radius of a bubble increases both because of proper motion and because of the expansion of the Universe

$$\int_{t'}^t dt'' \rightarrow \int_{t'}^t dt'' \frac{a(t)}{a(t'')}. \quad (20)$$

Thus, the extended volume in an expanding universe finally reads

$$\mathcal{V}_t^{\text{ext}}(t) = \frac{4\pi}{3} v_w^3 \int_{t_0}^t dt' \Gamma(t') \left(\frac{a(t')}{a(t)} \right)^3 \left[\int_{t'}^t dt'' \frac{a(t)}{a(t'')} \right]^3. \quad (21)$$

One would like to express Equation (21) as a function of temperature, given that dynamical quantities

relevant to phase transition processes in the early Universe are more conveniently computed as a function of temperature rather than time. A common assumption usually employed in the literature is the use of the MIT bag equation of state [46], under which the following equalities hold

$$\frac{dt}{dT} \stackrel{\text{bag}}{=} -\frac{1}{H(T)T}, \quad \frac{a(t_1)}{a(t_2)} \stackrel{\text{bag}}{=} \frac{T_2}{T_1}. \quad (22)$$

Moreover, the energy density of the Universe is often taken to be radiation dominated, setting the functional form for the Hubble expansion rate $H(T)$. These assumptions are valid for fast transitions in which the amount of vacuum energy released in the FOPT is subdominant with respect to the radiation energy density at the time. However, they do not necessarily hold for strong supercooled phase transitions, in which most of the Universe remains in the false vacuum during a period in which the temperature can drop significantly. Given that the radiation contribution scales as $\rho \propto T^4$ while the vacuum energy density $V(\phi_+, T) - V(\phi_-, T)$ generally increases for smaller temperatures (cf. e.g. Figures 4 and 5), strong supercooled phase transitions can exhibit periods of vacuum domination, or at least feature regimes where the vacuum and radiation contributions are comparable.

To properly account for these scenarios, ELENA employs a more general relation [47] between time and temperature that only assumes the adiabatic expansion of the Universe, and includes the contributions from both the Standard Model (SM) and BSM fields to the potential, in order to evolve the temperature over time

$$\frac{dT}{dt} = -3H(T) \frac{\partial_T V^T(\phi_+(T), T)}{\partial_{TT} V^T(\phi_+(T), T)}, \quad \frac{a(t_1)}{a(t_2)} = \exp \left(\int_{t_2}^{t_1} dt' H(t') \right), \quad (23)$$

where $\partial_T V = \partial V / \partial T$, $\partial_{TT} V = \partial^2 V / \partial T^2$ and V^T is meant to refer to the total potential, including the (T -dependent but ϕ -independent) SM component and the contribution from BSM fields. The expression used in ELENA for the computation of the extended volume thus reads

$$\mathcal{V}_t^{\text{ext}}(T) = \frac{4\pi}{3} v_w^3 \int_T^{T_c} dT' \frac{\Gamma(T')}{3H(T')} \frac{\partial_{TT} V^T(T')}{\partial_T V^T(T')} \left(\frac{a(T')}{a(T)} \right)^3 \left[\int_T^{T'} dT'' \frac{1}{3H(T'')} \frac{\partial_{TT} V^T(T'')}{\partial_T V^T(T'')} \frac{a(T)}{a(T'')} \right]^3, \quad (24)$$

with the ratio of scale factors given by

$$\frac{a(T_1)}{a(T_2)} = \exp \left(\int_{T_1}^{T_2} dT' \frac{1}{3} \frac{\partial_{TT} V^T(T')}{\partial_T V^T(T')} \right). \quad (25)$$

The actual computation is performed by the function `temperatures.compute_logP_f`, as exemplified in the Listing 4.

```
1 from temperatures import compute_logP_f
2
3 logP_f, Temps, ratio_V, Gamma, H = compute_logP_f(dp, V_min_value, S3overT, v_w = 1,
    units = 'MeV')
```

Listing 4: Computation of the fraction of Universe volume in false vacuum.

The function `temperatures.compute_logP_f` takes as required arguments the model class instance and two dictionaries, containing for each key temperature value the corresponding value of the potential

at the true minimum and the tunnelling action (`dp`, `V_min_value`, `S3overT` in the Listing 4), computed in the Listings 1 and 3 in our example. Optional arguments are the asymptotic wall velocity (`v_w`, set by default to 1) and the units for dimensionful quantities (`units`, default `'GeV'`). The function returns the natural logarithm of $P_f(T)$ as an array (`logP_f`), where each entry corresponds to its value at the temperature reported in the same entry in the array `Temps` (sorted in increasing values of temperature). For convenience, the function also returns the arrays `ratio_V`, `Gamma`, `H`, which contain the values of $\partial_{TT}V^T/\partial_TV^T(T)$, $\Gamma(T)$ and $H(T)$ for each temperature contained in `Temps`, so that they can be stored and used in other computations or consistency checks.

3.5 Computation of transition milestone temperatures

The knowledge of $P_f(T)$ allows for the computation of the temperatures at which important milestones in the transition are reached. We show in this section how ELENA can be used to compute the **nucleation**, **percolation** and **completion** temperatures, taking into account the full evolution of the transition.

The nucleation temperature T_n is defined as the moment when there is, on average, one nucleated bubble per Hubble volume. A commonly adopted heuristic criterion for estimating T_n in the literature is

$$\frac{\Gamma(T_n)}{H^4(T_n)} \equiv 1, \quad (26)$$

where Γ is the false vacuum decay rate Equation (12) and H is the Hubble parameter. For transitions at the electroweak scale, an even simpler estimation is often adopted ⁵, given by the condition $S_{E,3}(T_N)/T_N \sim 140$.

Despite being a commonly employed quantity to characterise the temperature at which a FOPT takes place, the nucleation temperature does not represent a relevant physical quantity in the characterisation of the FOPT dynamics; its extended usage in the literature has mainly been due to its simple numerical implementation, given that it does not require the determination of the false vacuum fraction. Indeed, it is possible to find realisations in which the phase transition takes place but the nucleation criterion is never met [47], particularly in the context of supercooled phase transitions. Nevertheless, we stress here that the nucleation temperature gives the correct estimate for the relevant transition temperature in the case of fast transitions.

ELENA provides the function `temperatures.N_bubblesH` to compute the average number of nucleated bubbles per Hubble volume as a function of temperature, by using the general equation [45]

$$N(t) = \frac{4\pi}{3} \int_{t_c}^t dt' \frac{\Gamma(T(t')) P_f(t')}{H^3(t')}, \quad (27)$$

from which the nucleation time t_n can be derived by looking for the solution $N(t_n) = 1$. The usage of `temperatures.N_bubblesH` is exemplified in the Listing 5.

```
1 from temperatures import compute_logP_f, N_bubblesH
2
```

⁵This is for example the default criterion in `CosmoTransitions`, see e.g. [19].

```
3 nH = N_bubblesH(Temps, Gamma, logP_f, H, ratio_V)
```

Listing 5: Computation of the average number of nucleated bubbles per Hubble volume.

The function `temperatures.N_bubblesH` takes as arguments a series of arrays `Temps`, `Gamma`, `logP_f`, `H`, `ratio_V`, corresponding to the output of the function `temperatures.compute_logP_f`, and returns an array containing the average number of bubbles per Hubble volume at each temperature value in `Temps`.

Percolation is defined as the moment when a connected cluster of bubbles that spans the entire Universe exists. The temperature at which it takes place characterises the peak frequency in the gravitational wave spectrum from a FOPT [45], given that at percolation the majority of bubbles responsible for the GW production have collided. It has been shown that a fully connected cluster of bubbles almost certainly forms when a specific value of the true vacuum fraction has been reached [48–50]; for cosmological phase transitions featuring uniformly nucleated spherical bubbles, this threshold value is $P_t \approx 0.29$ [51–53]. Therefore, the percolation temperature T_p is defined by the equation

$$P_f(T_p) = 0.71. \quad (28)$$

Percolation studies performed in the framework of solid state physics assume a static background, but in cosmological scenarios, the expansion of the Universe must be taken into account, especially in the case of slow supercooled phase transitions. In an expanding Universe, the expansion of bubbles towards each other competes with the expansion of space itself, which tends to increase the physical distance among bubbles. If the latter dominates, the bubbles might actually not collide, even if the fraction of the Universe in the false vacuum P_f decreases with time. Therefore, a necessary condition for the production of GW is that the physical volume in false vacuum, given by [54]

$$\mathcal{V}_{\text{phys}}(t) = a^3(t) P_f(t), \quad (29)$$

is decreasing when the bubbles are assumed to collide, which corresponds to

$$\frac{d\mathcal{V}_{\text{phys}}}{dt} = \mathcal{V}_{\text{phys}}(t) \left[\frac{d}{dt} \ln(P_f(t)) + 3H(t) \right] \leq 0. \quad (30)$$

We can track the evolution of $\mathcal{V}_{\text{phys}}(T)$ with temperature by substituting Equation (23) into Equation (30), from which one finds that the physical volume in the false vacuum is decreasing at a given temperature if the condition

$$\frac{d \ln P_f}{dT} \geq \frac{\partial_{TT} V^T}{\partial_T V^T} \quad (31)$$

is satisfied.

The percolation of bubbles in an expanding universe is questionable if $\mathcal{V}_{\text{phys}}$ is not decreasing at T_p [47, 54, 55], although it is worth noticing that percolation studies in an expanding spacetime have not been performed to this date [45], leaving open the question on how significantly the results can change compared to the static scenario.

Finally, the completion temperature is defined as the one at which the value of P_f has decreased below a certain threshold value, signalling that most of the Universe has been converted to the true

vacuum ⁶. It is given by

$$P_f(T_e) = \epsilon, \quad (32)$$

with $\epsilon \ll 1$. The value of T_e generally depends weakly on the choice of ϵ , except when the asymptotic limit of P_f is different from zero. A common choice is to require that at least 99% of the Universe has been converted to the true vacuum, $\epsilon = 0.01$.

The full computation of the transition milestones and verification of decreasing physical volume in the false vacuum at percolation in ELENA is exemplified in the Listing 6.

```

1 from utils import interpolation_narrow
2 from temperatures import compute_logP_f, N_bubblesH
3
4 is_physical = True
5
6 def is_increasing(arr):
7     return np.all(arr[:-1] <= arr[1:])
8
9 counter = 0
10 while counter <= 1:
11     if counter == 1:
12         temperatures = np.linspace(np.nanmax([T_min, 0.95 * T_completion]), np.nanmin(
13             [T_max, 1.05 * T_nuc]), n_points, endpoint = True)
14         action_vec(temperatures)
15         logP_f, Temps, ratio_V, Gamma, H = compute_logP_f(dp, V_min_value, S3overT, v_w =
16             1, units = units)
17         nH = N_bubblesH(Temps, Gamma, logP_f, H, ratio_V)
18         mask_nH = ~np.isnan(nH)
19         T_nuc = interpolation_narrow(np.log(nH[mask_nH]), Temps[mask_nH], 0)
20         mask_Pf = ~np.isnan(logP_f)
21         T_perc = interpolation_narrow(logP_f[mask_Pf], Temps[mask_Pf], np.log(0.71))
22         T_completion = interpolation_narrow(logP_f[mask_Pf], Temps[mask_Pf], np.log(0.01)
23         )
24
25         idx_compl = np.max([np.argmin(np.abs(Temps - T_completion)), 1])
26         test_completion = np.array([logP_f[idx_compl - 1], logP_f[idx_compl], logP_f[
27             idx_compl + 1]])
28         test_completion = test_completion[~np.isnan(test_completion)]
29
30         if not is_increasing(test_completion):
31             T_completion = np.nan
32         if counter == 1:
33             d_dT_logP_f = np.gradient(logP_f, Temps)
34             log_at_T_perc = interpolation_narrow(Temps, d_dT_logP_f, T_perc)
35             ratio_V_at_T_perc = interpolation_narrow(Temps, ratio_V, T_perc)
36             log_at_T_completion = interpolation_narrow(Temps, d_dT_logP_f, T_completion)
37             ratio_V_at_T_completion = interpolation_narrow(Temps, ratio_V, T_completion)

```

⁶We use the subscript e for “end”, to avoid confusion with the commonly used T_c notation that refers to the critical temperature.

```

34         if ratio_V_at_T_perc > log_at_T_perc:
35             is_physical = False
36         counter += 1

```

Listing 6: Computation of the transition temperatures milestones and of the evolution of physical volume in false vacuum.

The code in Listing 6 defines an auxiliary boolean flag `is_physical`, which will be set to `True` if the final SGWB prediction is reliable, and `False` if some underlying assumption in the calculation is violated (more details on these conditions are presented in the following). The function `is_increasing` simply verifies if a NumPy array is monotonically increasing, and is used as a further check on the completion of the transition. The `while` loop is implemented to improve the precision of the computation: the milestone temperatures are first calculated by employing the values of the action computed over the full range of temperatures between T_{min} and T_c ; once a first estimate of T_e and T_n is obtained, the action is computed over the temperature range $T \in [0.95 T_e, 1.05 T_n]$ with a smaller step size in temperature, to increase the precision of the integrations appearing in Equations (23), (25) and (27) over the most relevant temperature interval. Notice that this is an optional improvement: the user can skip this further computation and simply use the output from the first iteration, depending on the desired trade-off between computational time and precision. The rest of the code uses the already discussed functions `temperatures.compute_logP_f` and `temperatures.N_bubblesH` to compute $\ln(P_f(T))$ (`logP_f`) and $N(T)$ (`nH`), and some NumPy masks are defined to select only temperatures where the result of the computation is a float number (`mask_Pf` and `mask_nH`) to prevent potential issues in the subsequent computations due to the presence of NumPy Not a Number (NaN) (if any). The function `utils.interpolation_narrow` is a wrapper of `numpy.interp` that restricts the range of interpolation to avoid numerical problems that can result from the very large range of values that the functions $\ln(P_f(T))$ and $N(T)$ can span. It is called as `utils.interpolation_narrow(y, x, target)`, where `y` and `x` are NumPy arrays of the same dimension, and `y` is interpreted as a function of `x`; the function returns the value of the independent variable `x` for which the variable `y` is equal to `target`. If no such solution is found, it returns a `numpy.nan`. The code uses `utils.interpolation_narrow` to compute the nucleation (`T_nuc`), percolation (`T_perc`) and completion (`T_completion`) temperatures. The following lines isolate the range of $\ln(P_f(T))$ values immediately around the completion temperature (`test_completion`) and check if the value of $\ln(P_f(T))$ is decreasing at that temperature, to ensure that the fraction of volume in false vacuum is evolving towards smaller values when $P_f = \epsilon$. If this condition is not satisfied, the variable `T_completion` is set to `numpy.nan`, to indicate that a valid completion criterion was not found. Finally, the code computes $d \ln P_f / dT$ (`d_dT_logP_f`) over the full range of temperatures, as well as the values of $d \ln P_f / dT$ and $\partial_{TT} V^T / \partial_T V^T$ at percolation (`log_at_T_perc` and `ratio_V_at_T_perc`) and completion (`log_at_T_completion` and `ratio_V_at_T_completion`). In the case in which the condition in Equation (31) is violated, the flag `is_physical` is set to `False`.

We report in Figure 6 the results of the computation for the “Fast” benchmark in Table 2. The left panels show the value of the fraction of the Universe in false vacuum P_f (top) and of $-\ln P_f$ (bottom). Vertical lines signal the values of the nucleation (T_n , dashed line), percolation (T_p , dotted line) and completion (T_e , dot-dashed line) temperatures, while horizontal lines signal the values of $P_f = 0.71$

(dashed line) and $P_f = 0.01$ (dot-dashed line). On the right panels we report the evolution of the average number of bubbles per Hubble volume N , both on linear scale (top panel) and logarithmic scale (lower panel). Vertical lines follow the same convention as on the left panels, while the horizontal dashed line corresponds to the value $N = 1$. Figure 7 shows the same quantities as in Figure 6, but

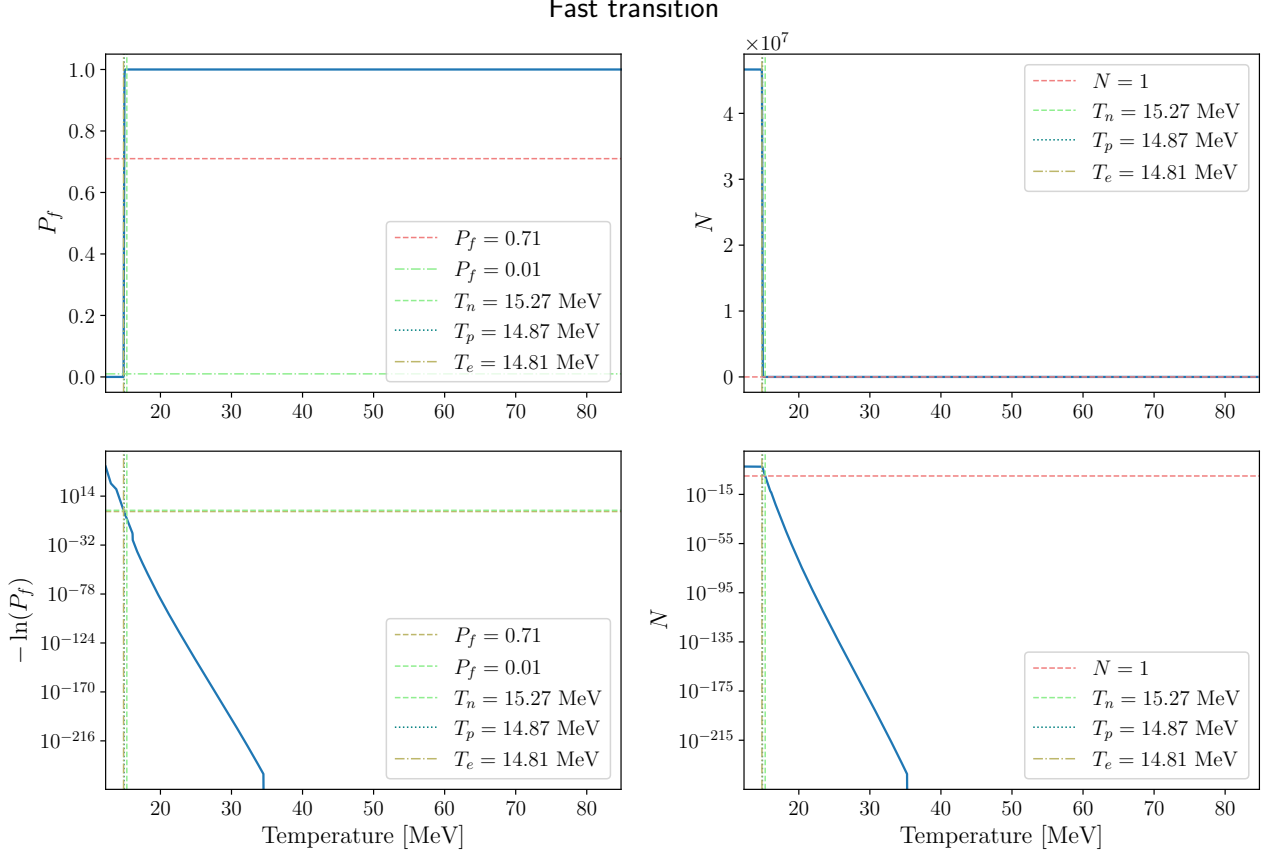


Figure 6: Evolution of the transition for the “Fast” benchmark point in Table 2. *Left panels* show the value of $P_f(T)$ (top) and $-\ln P_f(T)$ (bottom). *Right panels* show the evolution of $N(T)$ on linear (top) and logarithmic (bottom) scales. The vertical lines signal the values of the nucleation (T_n , dashed line), percolation (T_p , dotted line) and completion (T_e , dot-dashed line) temperatures. The horizontal lines on the left panels signal the values of $P_f = 0.71$ (dashed line) and $P_f = 0.01$ (dot-dashed line), while on the right panel they signal the value $N = 1$.

zoomed over the most interesting range of temperatures between nucleation and completion.

The same quantities are plotted in Figures 8 and 9 for the “Slow” benchmark in Table 2.

The most striking difference between the two benchmark points in Table 2 is obviously the duration of the transition: in the “Fast” transition, the nucleation of bubbles becomes sizeable at $T_n^{\text{fast}} = 15.27$ MeV, and the full transition completes shortly after, at $T_e^{\text{fast}} = 14.81$ MeV. Contrary, in the “Slow” transition the nucleation is sizeable at $T_n^{\text{slow}} = 13.94$ MeV, and the transition completes at $T_e^{\text{slow}} = 6.39$ MeV. Notice also the difference in the average number of bubbles per Hubble volume N , which reaches an asymptotic value of $N(T_{\min})^{\text{fast}} = 4.67 \times 10^7$ in the “Fast” benchmark, and $N(T_{\min})^{\text{slow}} = 48.73$ in the “Slow” one. This generally translates into larger SGWB signals from slow transitions, given that the colliding bubbles are larger and thus contain a larger amount of energy to produce GW.

The evolution of the physical volume of the Universe in the false vacuum is reported in Figure 10 for the

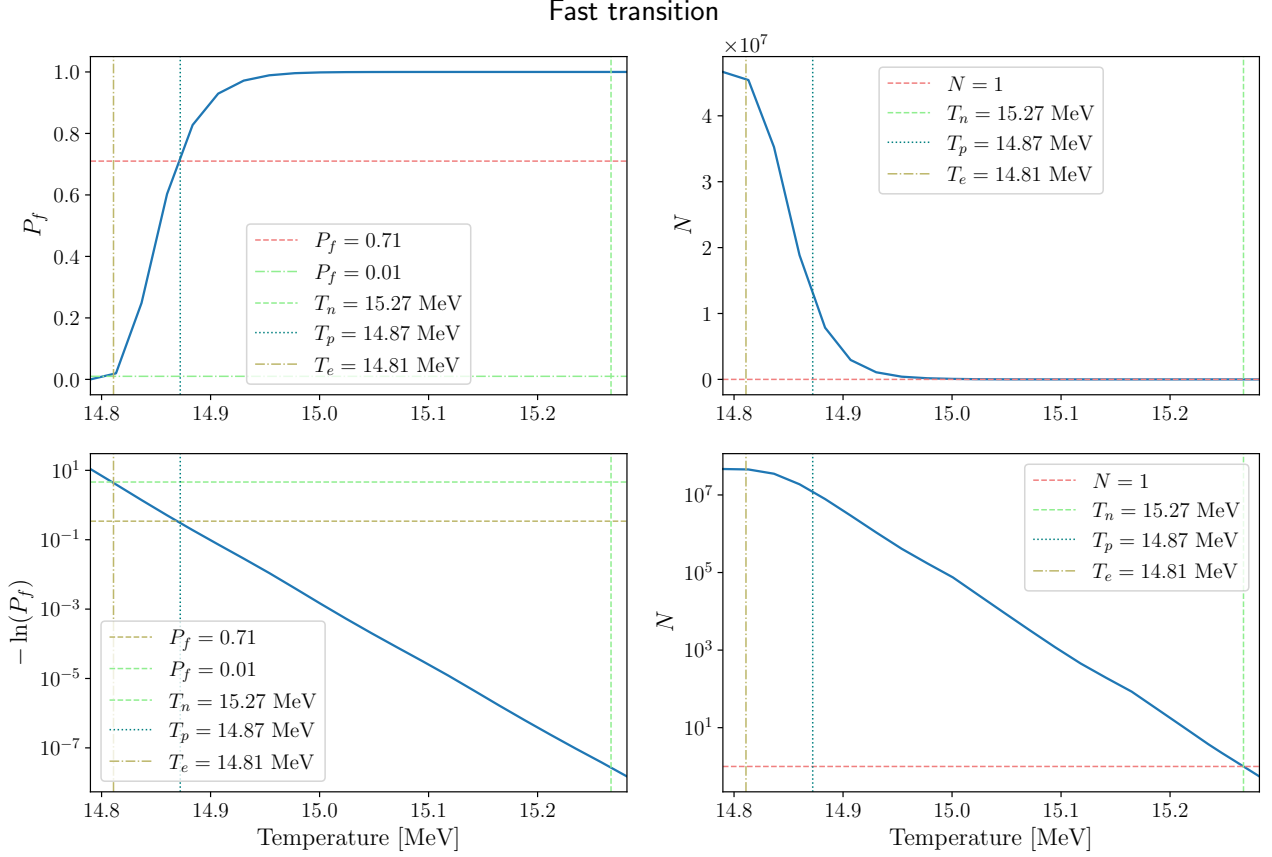


Figure 7: Same as Figure 6, but zoomed over the range of temperatures between nucleation and completion.

"Fast" transition, and in Figure 11 for the "Slow" one. In each figure the dashed curves represent the value of $\partial_{TT} V^T / \partial_T V^T$, while the continuous curves represent $d \ln P_f / dT$; the temperature values where the condition in Equation (31) is satisfied (and thus where the physical volume in the false vacuum is decreasing) have a light blue background, while temperature values for which $\mathcal{V}_{\text{phys}}$ is increasing have a light coral background. For reference, the values of the milestone temperatures are reported as vertical dashed (T_n), dotted (T_p) and dot-dashed (T_e) lines. The left panels show the evolution of $\mathcal{V}_{\text{phys}}$ over the full range of temperatures $T \in [T_{\min}, T_c]$, while the right panels are zoomed over $T \in [T_e, T_n]$. It can be noticed that, for both benchmarks, $\mathcal{V}_{\text{phys}}$ is actually increasing at T_n , providing further evidence that the use of the nucleation temperature as the characteristic temperature for the computation of the SGWB spectrum should be avoided. In the "Fast" benchmark point $d \ln P_f / dT$ rapidly increases at low temperatures with respect to $\partial_{TT} V^T / \partial_T V^T$, signalling a fast conversion of the physical volume from false to true vacuum state. In the "Slow" transition, the value of $\mathcal{V}_{\text{phys}}$ varies more gently, as evidenced by the smaller hierarchy of values between $d \ln P_f / dT$ and $\partial_{TT} V^T / \partial_T V^T$ when the physical volume in false vacuum is decreasing, as well as by the extended range of temperatures over which the transition takes place. For both the "Fast" and "Slow" benchmark points in Table 2, the condition Equation (31) is satisfied at the percolation temperature. Nevertheless, it could also be the case that $\mathcal{V}_{\text{phys}}$ decreases at T_p but increases afterwards, which makes it fundamental to also check the completion condition from Equation (32).

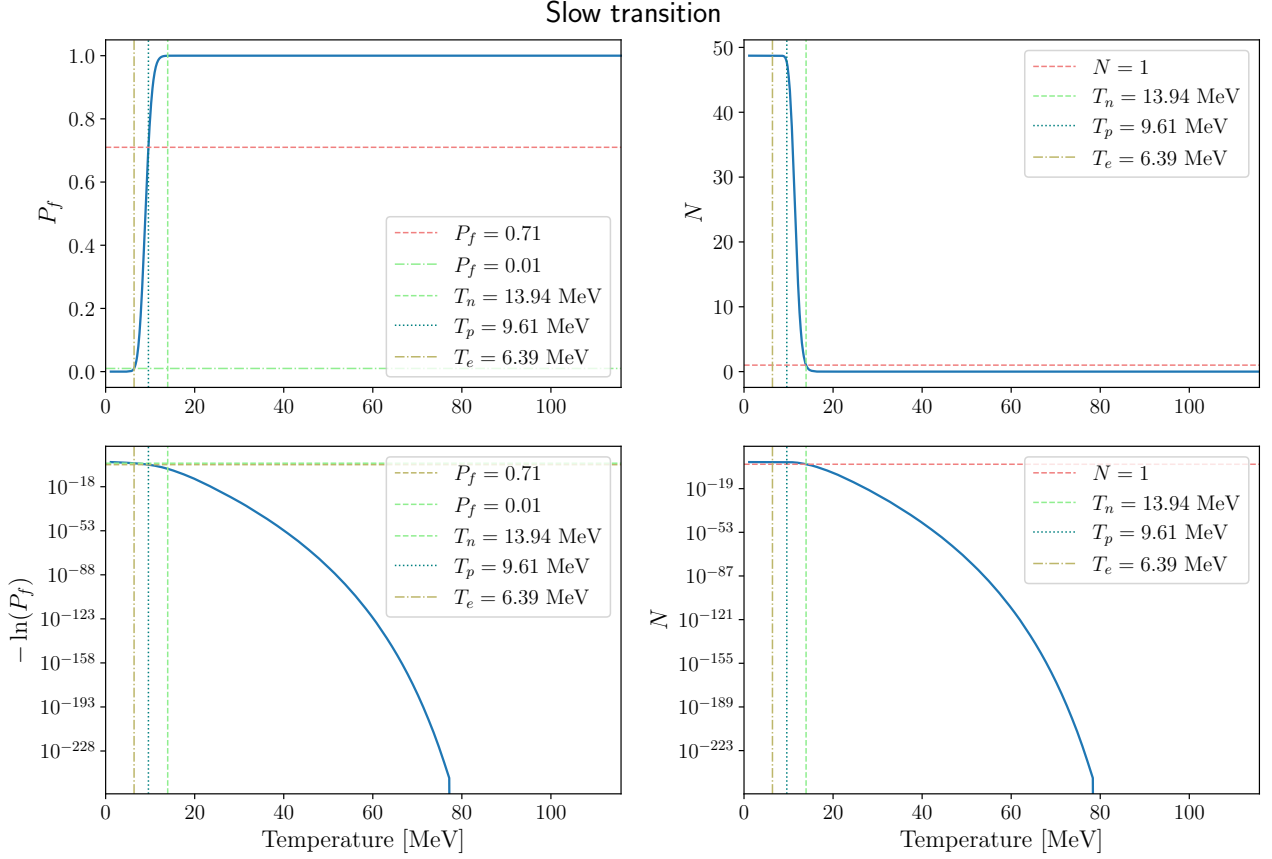


Figure 8: Same as in Figure 6, but for the “Slow” benchmark in Table 2.

3.6 Computation of thermal parameters determining the GW production

The production of GW sourced by a FOPT mainly depends on three thermal parameters characterising the transition: the reheating temperature T_{RH} , the strength of the transition α and the mean bubble separation at percolation R_* [45]⁷. In addition, the efficiency parameters α_∞ [58] and α_{eq} [59] determine the fraction of vacuum energy that is converted into GWs, as well as the distribution of this energy fraction into the different sources of GWs (bubble walls collisions, sound waves and turbulence in the plasma). Finally, the transition strength and the spectral shape of the GWs produced by sound waves and turbulence depend as well on the speed of sound in the plasma, which in the presence of a non-negligible vacuum energy differs from the ideal relativistic gas assumed in the bag model.

ELENA implements general formulae for the computation of these quantities, going beyond common assumptions usually found in the literature. We exemplify how to compute the thermal parameters of a FOPT and review their expressions in the following subsections.

⁷Note that the characteristic length-scale given by R_* is sometimes traded for a characteristic duration of the FOPT β^{-1} , especially when computing gravitational waves generated by bubble collisions and relativistic shells [56, 57]. The parameter β can be related to R_* through $\beta = (8\pi)^{1/3} v_w / R_*$, assuming the nucleation rate to follow $\Gamma \sim \Gamma_n e^{\beta t}$. Notice that slow transitions can significantly deviate from this assumption, as shown later in Section 3.6.2.

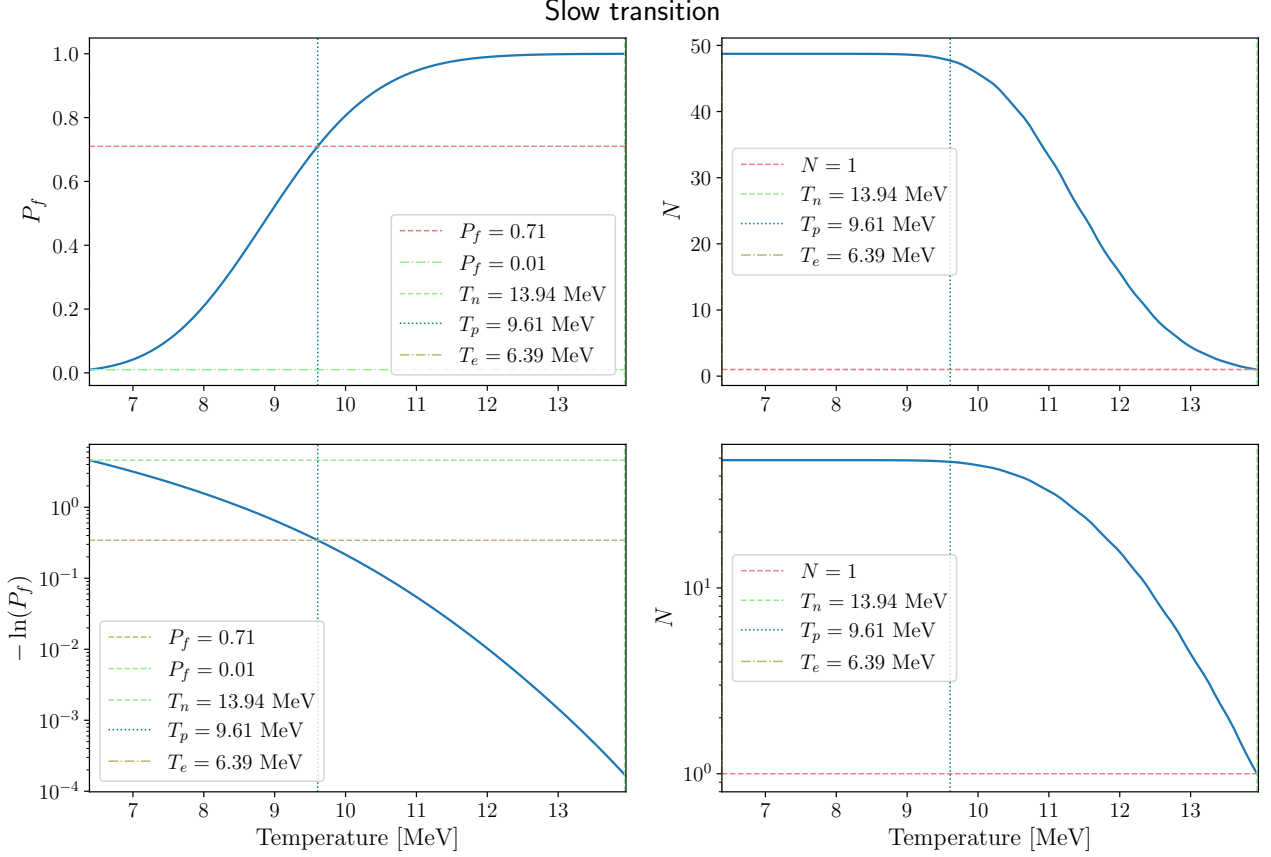


Figure 9: Same as Figure 8, but zoomed over the range of temperatures between nucleation and completion.

3.6.1 Transition strength α and efficiency parameters α_∞ , α_{eq}

Several definitions exist for the computation of the transition strength α (cf. [45] for a detailed analysis). ELENA adopts the general definition [60, 61]

$$\alpha = \frac{4}{3} \frac{\bar{\theta}_f(T_p) - \bar{\theta}_t(T_p)}{w_f(T_p)}, \quad (33)$$

where $w = -T(\partial V^T/\partial T)$ is the enthalpy density of the full system (Standard Model plus new fields), $\bar{\theta} = (\rho - 3p/c_{s,t}^2)/4$ is the pseudotrace, with $c_{s,t}^2(T)$ the speed of sound in the plasma in the broken phase, ρ the energy density and p the pressure. The pseudotrace is a generalisation of the vacuum energy. The latter should only be used when the assumptions of the MIT bag model are fulfilled [45]. In Equation (33), the subscript $f(t)$ indicates the value of these quantities in the false (true) minimum. This definition of α does not assume the validity of the bag model, preventing the appearance of large errors in the estimation of the kinetic energy fraction [60, 61].

The value of α determines the total amount of energy that is released in the FOPT. The fraction of energy that is converted into each GW source depends on the efficiency parameter α_∞ [58]

$$\alpha_\infty = \frac{1}{18} \frac{\Delta m^2 T_p^2}{w_f(T_p)}, \quad \text{with} \quad \Delta m^2 = \sum_i c_i n_i (m_{i,t}^2 - m_{i,f}^2), \quad (34)$$

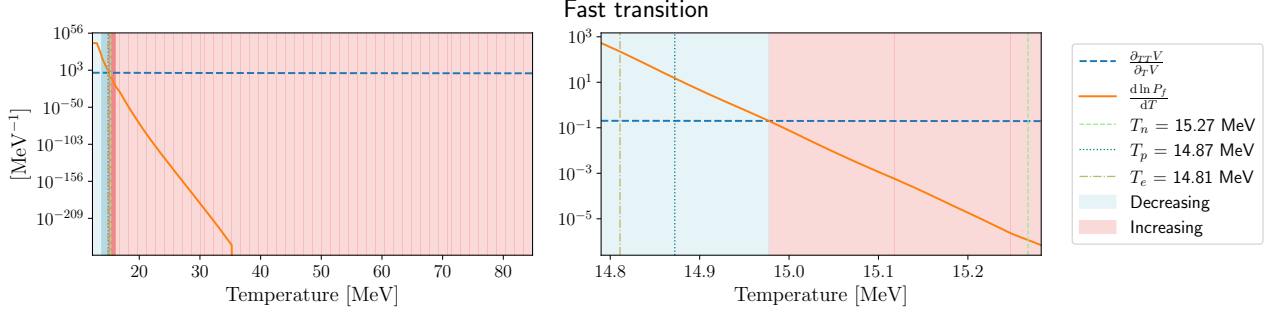


Figure 10: The evolution of $d \ln P_f / dT$ (continuous curve) and $\partial_{TT} V^T / \partial_T V^T$ (dashed curve) with temperature for the “Fast” benchmark in Table 2. Temperature values where the condition in Equation (31) is satisfied (violated) are highlighted by a light-blue (light-coral) background. The vertical dashed, dotted and dot-dashed lines represent the values of the nucleation (T_n), percolation (T_p) and completion (T_e) temperatures, respectively. The left panel shows the entire range of temperatures over the full transition duration, the right panel is zoomed over the temperature values between completion and nucleation.

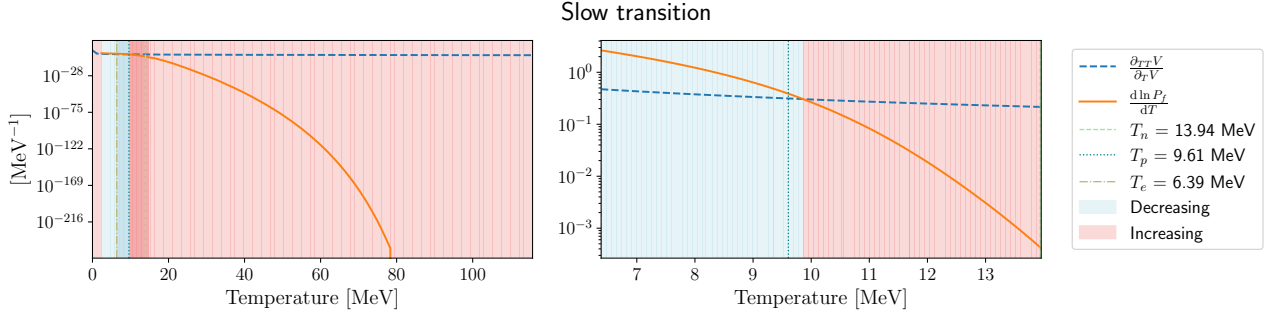


Figure 11: Same as in Figure 10 but for the “Slow” point in Table 2.

where the sum runs over all species gaining a mass, n_i are the internal degrees of freedom and $c_i = 1$ ($1/2$) for bosons (fermions). The parameter α_∞ represents the minimal transition strength required to fully overcome the leading-order (LO) friction term of the plasma, which results from the fields becoming massive due to the symmetry breaking: if $\alpha > \alpha_\infty$, the excess energy goes into accelerating the bubble walls, resulting into bubble-wall velocities $v_w \rightarrow 1$. In the opposite scenario, the bubble walls reach a terminal velocity $v_w < 1$, whose exact determination requires a dedicated analysis [58]. If the bubble walls reach relativistic velocities, next-to-leading order (NLO) friction terms, which are proportional to the wall Lorentz factor γ_w , become relevant. The effect of the NLO friction is instead quantified by the parameter α_{eq} [62–64], given by

$$\alpha_{\text{eq}} = \frac{4g^2 \Delta m_V T_p^3}{3w_f(T_p)}, \quad \text{with} \quad g^2 \Delta m_V = \sum_X 3g_X^2 (m_{X,t} - m_{X,f}). \quad (35)$$

The sum in Equation (35) only runs over the gauge bosons that become massive during the transition. The quantity α_{eq} is related to the terminal velocity that the bubble walls reach due to NLO friction terms [63, 64]

$$\gamma_{\text{eq}} = \frac{\alpha - \alpha_\infty}{\alpha_{\text{eq}}}. \quad (36)$$

γ_{eq} needs to be compared to the Lorentz factor that the wall would reach in the absence of the NLO pressure [59]

$$\gamma_* \equiv \frac{2 R_*}{3 R_0}, \quad (37)$$

where the parameter R_0 is defined in Ref. [59]. Strictly speaking, the quantity appearing in the numerator of Equation (37) should be the mean bubble radius. However, the mean bubble separation and mean bubble radius are comparable at percolation, and R_* must in any case be computed to determine the SGWB spectrum, cf. Section 3.6.2. Thus, we can trade the two quantities for the sake of numerical efficiency, provided Equation (37) is only evaluated at the percolation temperature. Similarly to the LO case, if $\gamma_* \leq \gamma_{\text{eq}}$, all the vacuum energy is then converted into accelerating the bubble walls, while if $\gamma_* > \gamma_{\text{eq}}$, the bubble walls reach a terminal velocity and the exceeding energy shall be released into the surrounding plasma, generating GWs via sound waves and turbulence.

The quantities discussed in this subsection need to be derived only at the percolation temperature, which represents the moment when most of the GWs are generated. For completeness, we show in the Listing 7 how to evaluate them in ELENA at any generic temperature $T \in [T_{\text{min}}, T_c]$ where the function `action_over_T(T)` from the Listing 3 has already been executed. The user can easily adapt the example presented in the Listing 7 to different models or computation needs.

```

1 from GWparams import alpha_th_bar # This is the definition of \alpha
2
3 def c_alpha_inf(T, units):
4     v_true = true_vev[T]
5     v_false = false_vev[T]
6     Dm2_photon = 3 * g**2 * (v_true**2 - v_false**2)
7     Dm2_scalar = 3 * lambda_ * (v_true**2 - v_false**2)
8     numerator = (Dm2_photon + Dm2_scalar) * T**2 / 24
9     rho_tot = - T * 3 * (dp.dVdT(v_false, T, include_radiation=True, include_SM =
10     True, units = units) ) / 4
11     rho_DS = - T * 3 * (dp.dVdT(v_false, T, include_radiation=True, include_SM =
12     False, units = units) ) / 4
13     return numerator / rho_tot, numerator / rho_DS
14
15 def c_alpha_eq(T, units):
16     v_true = true_vev[T]
17     v_false = false_vev[T]
18     numerator = (g**2 * 3 * (g * (v_true - v_false)) * T**3)
19     rho_tot = - T * 3 * (dp.dVdT(v_false, T, include_radiation=True, include_SM =
20     True, units = units) ) / 4
21     rho_DS = - T * 3 * (dp.dVdT(v_false, T, include_radiation=True, include_SM =
22     False, units = units) ) / 4
23     return numerator / rho_tot, numerator / rho_DS
24
25 action_over_T(T_perc)
26 alpha, alpha_DS = alpha_th_bar(T_perc, dp, V_min_value, false_vev, true_vev, units =
27     units)
28 alpha_inf, alpha_inf_DS = c_alpha_inf(T_perc, units)

```

```

24 alpha_eq, alpha_eq_DS = c_alpha_eq(T_perc, units)
25
26 gamma_eq = (alpha - alpha_inf) / alpha_eq
27
28 if alpha < alpha_inf: # this implies v_w < 1
29     is_physical = False
30
31 if gamma_eq < 1 / np.sqrt(1 - 0.99**2): # this would imply v_w < 0.99
32     is_physical = False

```

Listing 7: Computation of the transition strength and of the efficiency parameters.

ELENA provides the function `GWparams.alpha_th_bar` to compute the transition strength at a given temperature using the Equation (33). The definitions of α_∞ and α_{eq} shall be provided by the user following the specific model under analysis; we provide in the Listing 7 the code to implement these computations for the model in Equation (10), defining the functions `c_alpha_inf` and `c_alpha_eq`, respectively. Notice that, while the masses appearing in Equations (34) and (35) generally have a vacuum component (function of the scalar vevs) and a thermal component (function of the temperature), the latter one cancels in the differences as long as the temperature on both sides of the bubble wall is the same. The functions `GWparams.alpha_th_bar`, `c_alpha_inf` and `c_alpha_eq` return each a tuple of two elements: the first one is the result of Equation (33), while the second one is the transition strength normalised to the enthalpy density of the non-Standard Model fields only, i.e. $w_{\text{BSM}} = -T(\partial V^{\text{BSM}}/\partial T)$. These further efficiency factors are relevant to the transition dynamics in scenarios where the new fields are effectively decoupled from the SM, forming a secluded dark sector [21].

The functions `c_alpha_inf` and `c_alpha_eq` in the Listing 7 simply take as argument the temperature `T`, expressed in the appropriate `units`. The function `GWparams.alpha_th_bar` takes as arguments the temperature (`T_perc` in this example), the model class (`dp`) and three dictionaries `V_min_value`, `false_vev`, and `true_vev`, each containing for the key temperature the values of the free energy, and the location of the false and true vacua, respectively. The line `action_over_T(T_perc)` computes these quantities at the temperature `T_perc` (cf. Listing 3). The following lines compute the transition strength normalised to the total enthalpy density (`alpha`) as well as to the BSM contribution only (`alpha_DS`), the LO (`alpha_inf`, `alpha_inf_DS`) and NLO efficiency factors (`alpha_eq`, `alpha_eq_DS`) with analogous normalisation conventions. `gamma_eq` is the resulting terminal bubble wall Lorentz factor, Equation (36). Some discussion is in order concerning the last block of code in Listing 7: here the boolean flag `is_physical` is set to `False` if the condition $\alpha > \alpha_\infty$ is not met, or if $\gamma_{\text{eq}} \lesssim 7$ (corresponding to a wall velocity $v_w < 0.99$). Even though these solutions are perfectly viable from a FOPT point of view, the computation of $P_f(T)$ previously performed is not reliable in these cases, given that the argument `v_w` was set to 1 in `temperatures.compute_logP_f` (cf. Listing 4). As previously mentioned, in the case $\alpha < \alpha_\infty$, a dedicated analysis is necessary to estimate the terminal bubble wall velocity v_w . ELENA can compute the transition history for generic values of the bubble wall velocity by setting the appropriate argument `v_w`, but the physical value of $v_w < 1$ shall be provided by the user. Since in the example code provided here we assumed $v_w = 1$, the flag `is_physical` is then set to `False` if the transition is not strong enough to reach $v_w \rightarrow 1$, signalling a breaking of the underlying assumptions and the need to explicitly compute

the value of v_w . This should however not be interpreted as an exclusion of the considered benchmark point.

We report the evolution of α (continuous curves), α_∞ (dashed curves) and α_{eq} (dot-dashed curves) as functions of temperature in Figure 12 (Figure 13) for the “Fast” (“Slow”) benchmark in Table 2. Left panels show the full temperature range $T \in [T_{\min}, T_c]$, while right panels are zoomed over $T \in [T_e, T_n]$. For reference, the nucleation (T_n), percolation (T_p) and completion (T_e) temperatures are signalled as vertical dashed, dotted and dot-dashed lines, respectively. We notice that in both benchmark points the condition $\alpha > \alpha_\infty$ is satisfied at T_n , which is when bubble nucleation starts to be relevant, thus justifying the assumption on the bubble walls velocity $v_w = 1$ in the computation of P_f .

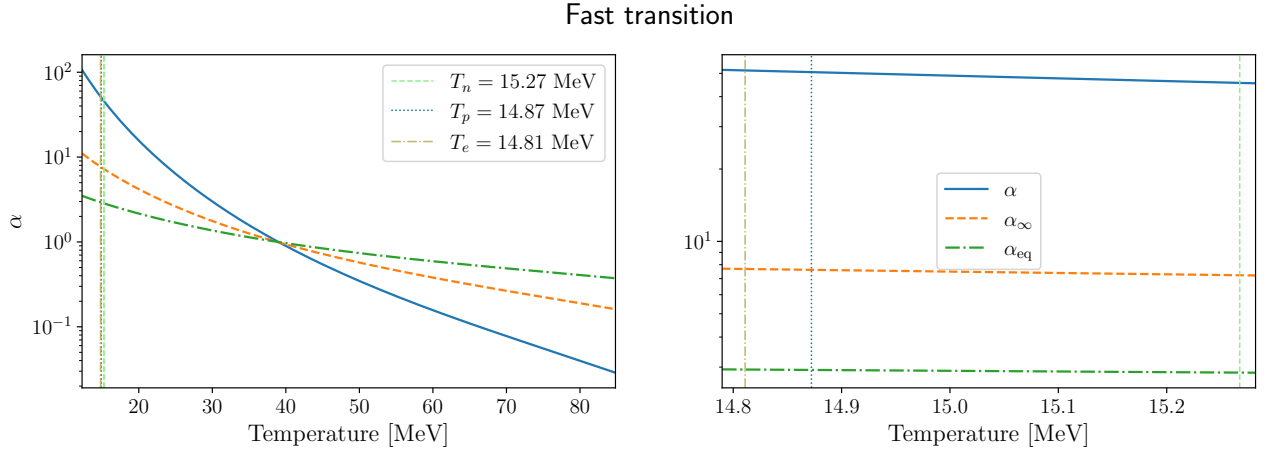


Figure 12: Evolution of the available transition strength α (continuous curve), LO friction term α_∞ (dashed curve) and NLO friction term (dot-dashed curve) as function of temperature for the “Fast” benchmark point in Table 2. The nucleation (T_n), percolation (T_p) and completion (T_e) temperatures are signalled as vertical dashed, dotted and dot-dashed lines, respectively. The left panel shows the full temperature range over the transition evolution, while the right panel zooms on the interval between T_e and T_n .

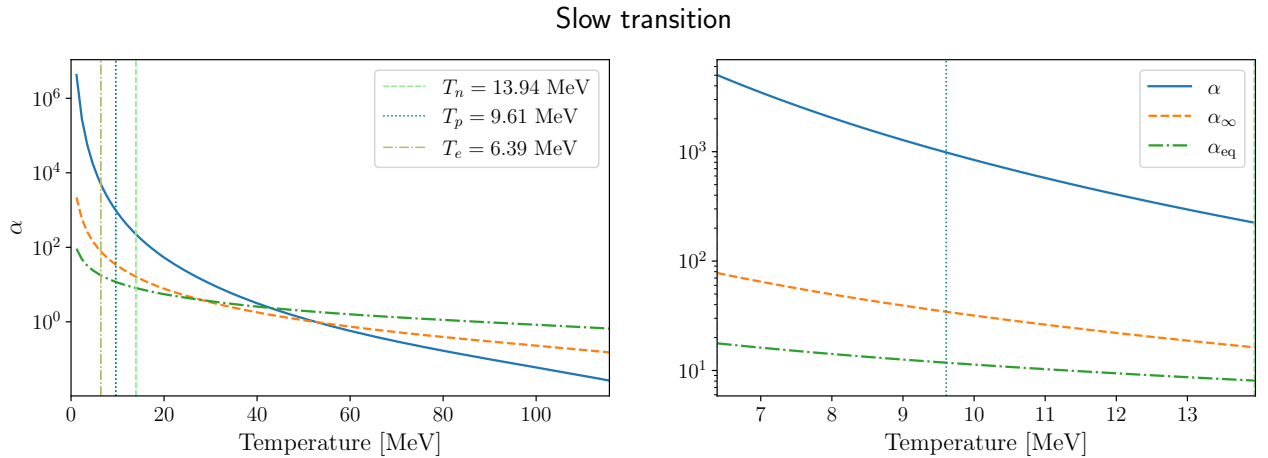


Figure 13: Same as in Figure 12 but for the “Slow” benchmark in Table 2.

3.6.2 Mean bubble separation and transition duration

The amplitude and peak frequency of the SGWB produced by a FOPT strongly depend on the characteristic length-scale of the transition, which can be specified using different parameters, requiring to choose one among them [45]. The parameter used as input in many numerical hydrodynamic simulations that provide fits to the SGWB spectrum (cf. e.g. [65]) is the mean bubble separation

$$R_{\text{sep}}(t) = n(t)^{-\frac{1}{3}}, \quad (38)$$

where n is the bubble number density [43] given by

$$n(t) = \int_{t_c}^t dt' \Gamma(t') P_f(t') \frac{a^3(t')}{a^3(t)}. \quad (39)$$

Given that the use of R_{sep} is recommended over other possible length-scale choices [45], ELENA implements the computation of the mean bubble separation with the function `temperatures.R_sepH`, which returns a tuple of two floats, $R_{\text{sep}}(T)H(T)$ and R_{sep} . The first quantity normalises the mean bubble separation to the Hubble radius, and is the one inputted in SGWB fits; the second quantity is the dimensionful physical distance. The use of `temperatures.R_sepH` is exemplified in the Listing 8.

```
1 from temperatures import R_sepH
2
3 RH, R = R_sepH(Temps, Gamma, logP_f, H, ratio_V)
4 RH_interp = interpolation_narrow(Temps, RH, T_perc)
```

Listing 8: Computation of the mean bubble separation.

The function `temperatures.R_sepH` requires as arguments the outputs of `temperatures.compute_logP_f`, see the Listing 4 and subsequent discussion. The arrays `RH` and `R` contain the value of $R_{\text{sep}}(T)H(T)$ and $R_{\text{sep}}(T)$ for each temperature value in the array `Temps`. In the last line of Listing 8, the function `utils.interpolation_narrow` is used to find the value of the mean bubble separation at the percolation temperature, `RH_interp`. We report as a continuous curve in Figure 14 (Figure 15) the evolution of $R_{\text{sep}}(T)H(T)$ for the “Fast” (“Slow”) benchmark point in Table 2. The temperatures of nucleation (T_n), percolation (T_p) and completion (T_e) are reported as vertical dashed, dotted and dot-dashed lines, respectively. The horizontal dashed line represents the mean bubble separation at percolation $R_{\text{sep}}(T_p)H(T_p) = R_{\text{sep}*}H_*$, which is the parameter entering into the SGWB spectrum. Left panels show the full range of transition temperatures, while right panels provide a zoomed view over the temperature range between nucleation and completion. For the “Fast” transition in Figure 14 the nucleation of bubbles is sizeable until the completion temperature T_e , thus resulting in a relatively small value of the mean bubble separation at percolation in units of the Hubble radius, $(R_{\text{sep}*}H_*)^{\text{fast}} = 7.1 \times 10^{-3}$. For the “Slow” transition in Figure 15, on the other hand, the nucleation of new bubbles becomes ineffective before the percolation temperature T_p , and the conversion of volume from false to true vacuum mainly proceeds via the expansion of already nucleated bubbles. This results in a slow increase in $R_{\text{sep}}(T)H(T)$ due to the expansion of the Universe, as expected from Equation (39), even though the physical volume in the false vacuum is decreasing due to the bubbles walls expansion,

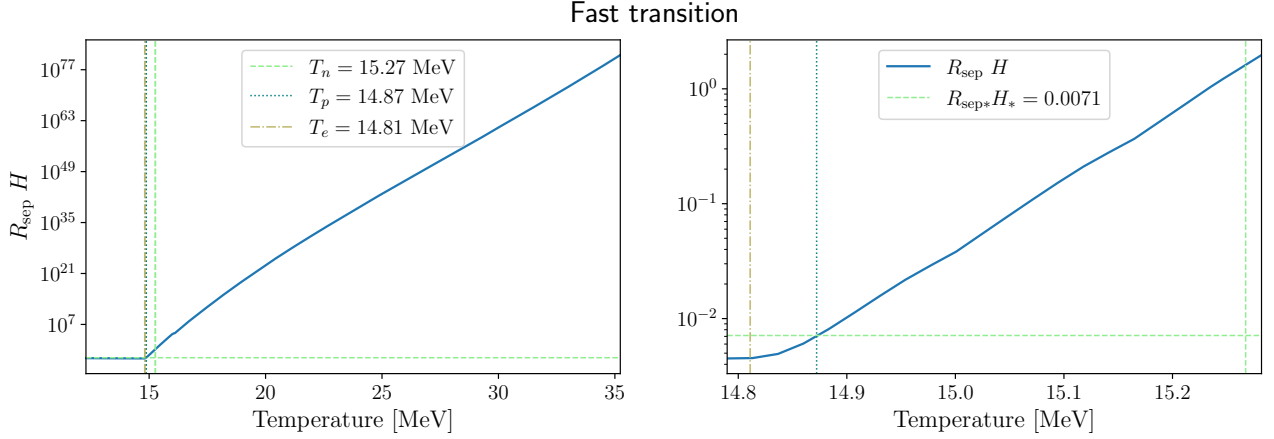


Figure 14: Evolution of the mean bubble separation in units of the Hubble radius, $R_{\text{sep}}(T)H(T)$, as a function of temperature for the “Fast” benchmark point in Table 2. Vertical lines represent the nucleation (T_n), percolation (T_p) and completion (T_e) temperatures as dashed, dotted and dot-dashed lines, respectively. The horizontal dashed line represents the value of the mean bubble separation at percolation $R_{\text{sep}}(T_p)H(T_p) = R_{\text{sep}*}H_*$. The left panel shows the evolution over the full range of temperatures, the right panel zooms on the interval between T_n and T_e .

cf. Figure 11. The resulting mean bubble separation at percolation in units of Hubble radius is thus sizeable, $(R_{\text{sep}*}H_*)^{\text{slow}} = 0.53$.

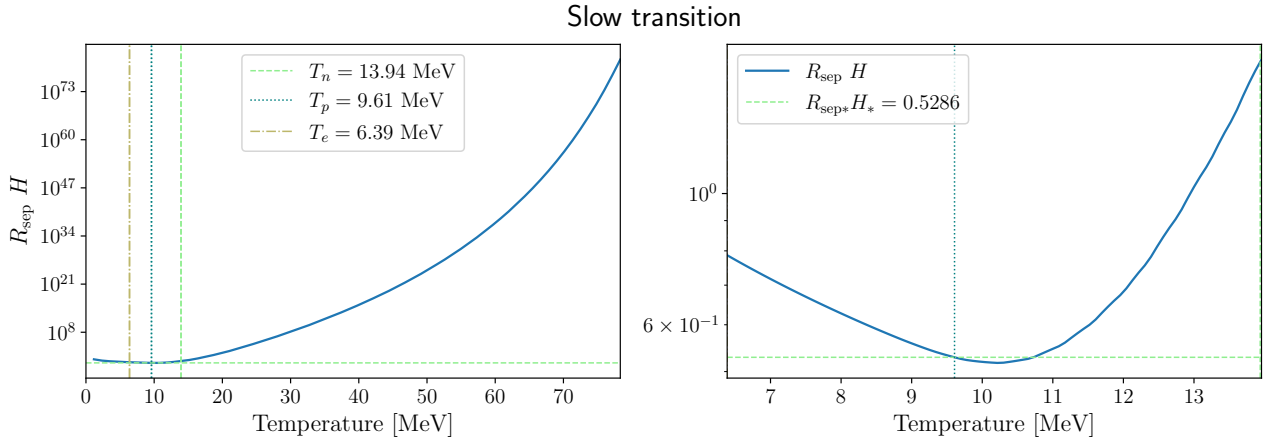


Figure 15: Same as in Figure 14, but for the “Slow” benchmark point in Table 2.

An alternative to R_{sep} widely used in the literature is the inverse time-scale of the transition, β [45]. This quantity is related to the shape of the tunnelling action $S_{E,d}$ when it can be Taylor expanded around the nucleation time and truncated to the linear term as

$$\beta_* = - \left. \frac{dS}{dt} \right|_{t_*}. \quad (40)$$

This computation thus assumes an exponential nucleation rate, $\Gamma(t) = \Gamma(t_*) \exp(\beta_*(t - t_*))$. However, an exponential nucleation rate is characteristic of fast transitions only, while using this parametrisation for slow transitions can give wrong results. We show this for our benchmark points in Table 2 in the following, by employing the more general parametrisation for the nucleation rate that includes the

quadratic term in the Taylor expansion

$$\Gamma(t) = \Gamma_n \exp \left[\beta(t - t_n) - \frac{1}{2} \gamma^2 (t - t_n)^2 \right], \quad (41)$$

where t_n is the nucleation time and $\Gamma_n = \Gamma(t_n)$ is the false vacuum decay rate at nucleation. Time can be related to temperature by

$$t - t_0 = \int_{t_0}^t dt = \int_{T(t)}^{T(t_0)} \frac{dT'}{3H(T')} \frac{\partial_{TT} V^T(T')}{\partial_T V^T(T')}. \quad (42)$$

ELENA provides the function `GWparams.beta` to readily extract the values of the β and γ parameters in Equation (41), by simply using as arguments the output of `temperatures.compute_logP_f` (cf. Listing 6), a starting (`T_nuc`) and final (`T_perc`) temperatures, as well as an optional argument `verbose` (default `False`) that controls its output. To avoid the appearance of dimensionful quantities that can be very small and result in numerical instabilities, `GWparams.beta` normalises time in units of the Hubble expansion rate at nucleation H_n , thus employing $H_n(t - t_n)$ as the independent variable, and extracting β/H_n and γ/H_n as coefficients. The usage of `GWparams.beta` is demonstrated in the Listing 9.

```
1 from GWparams import beta
2
3 beta_Hn, gamma_Hn, times, Gamma_t, Temps_t, H_t = beta(Temps, ratio_V, Gamma, H,
    T_nuc, T_perc, verbose = True)
```

Listing 9: Computation of the nucleation rate coefficients.

The function `GWparams.beta` performs a fit of the numerically computed decay rate $\Gamma(t)$ to the expression in Equation (41), setting t_n as the time of the starting temperature, and considering a time interval extended until the final temperature. If `verbose = False` the function only returns the values of β/H_n (`beta_Hn`) and γ/H_n (`gamma_Hn`). If `verbose = True` the function also returns an ensemble of arrays, containing in each element the corresponding values of $H_n \times (t - t_n)$ (`times`), $\Gamma(t - t_n)$ (`Gamma_t`), $T(t - t_n)$ (`Temps_t`) and $H(t - t_n)$ (`H_t`), where the $Q(t - t_n)$ notation indicates that the Q quantity is evaluated at different $(t - t_n)$ time values⁸.

The results of the numerical fit to Equation (41) for the “Fast” point in Table 2 are reported in Figure 16 for the evolution between the nucleation and percolation times. The left panel in the figure shows the evolution of the temperature with time, from which it can be noted that percolation is reached after 2.6×10^{-2} unit times $1/H_n$. On the right panel we report the numerical computation of the nucleation rate (continuous line) and the expression computed by inserting the fit results (reported on the right-side of the figure) into Equation (41). Clearly, the exponential nucleation is a very good approximation for this point, with the ratio $\gamma/\beta = 0.04$ indicating that the second-order term in the Taylor expansion can be safely neglected. In this case, using the β parameter to characterise the transition is justified. Analogously, we report in Figure 17 the results of the numerical fit to the “Slow” benchmark point in Table 2. As can be noted in the left panel and as expected, the duration of this transition is much

⁸We write instead $H_n \times (t - t_n)$ to stress that this is a multiplication between the constant quantity H_n and the time variable $t - t_n$.

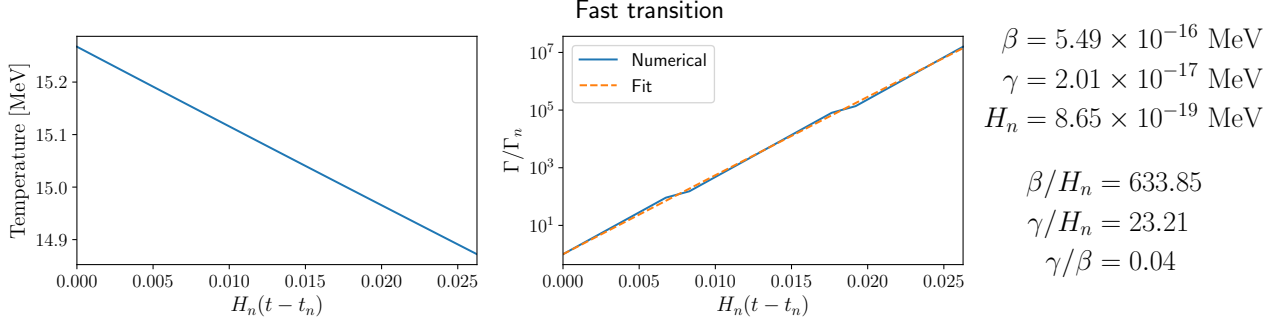


Figure 16: Nucleation evolution for the “Fast” point in Table 2. The left panel shows the evolution of temperature as a function of time between nucleation and percolation, with the time expressed in units of the Hubble time at nucleation $1/H_n$. The right panel shows the nucleation rate computed numerically in ELENA (continuous line) and the result of Equation (41) with the numerical parameters reported on the right side of the figure (dashed line).

longer than the “Fast” one, taking $0.37/H_n$ from nucleation to percolation. More importantly, the results of the numerical fit reported on the right-side of the figure show that the second order term in the Taylor expansion is relevant, with a ratio $\gamma/\beta = 0.46$. Thus, this benchmark point cannot be approximated with an exponential nucleation, showing explicitly that the employment of β to characterise the SGWB spectrum would produce unreliable results. As exemplified by the discussion

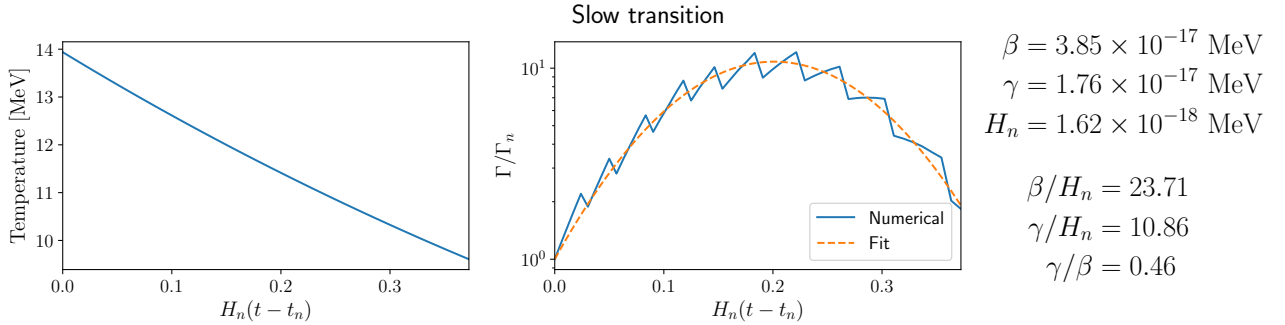


Figure 17: Same as in Figure 16 but for the “Slow” point in Table 2.

in this Section, it is strongly advised to employ R_{sep} to characterise the typical length-scale of the transition [45]. This choice is followed in ELENA, which provides the function `temperatures.R_seph` for the computation of R_{sep} ; the user can nevertheless easily compute the β and γ parameters using `GWparams.beta`, if these quantities are of interest.

3.7 Computation of the gravitational waves spectrum

Once the thermal parameters of the FOPT are known, the SGWB can be finally computed. Among the different possibilities present in the literature, ELENA employs the relations provided in Ref. [59], which use the mean bubble separation at percolation $R_{\text{sep}}*H_*$ as input parameter (instead of the more commonly used inverse time-scale β) and that can thus also apply to scenarios where the nucleation rate is not exponential (cf. discussion in Section 3.6.2).

The GW spectra from Ref. [59] are implemented in the class `GWparams.GW_SuperCooled`. The user can

create an instance of the class by inputting the thermal parameters and then access the predicted GW spectra at any given frequency using the class methods, as exemplified in Listing 10.

```

1 from GWparams import GW_SuperCooled
2
3 inst = GW_SuperCooled(T_perc, alpha, alpha_inf, alpha_eq, R_star, gamma_star, H_star,
4                       c_s = np.sqrt(c_s2), v_w = 1, units = units, dark_dof = 4)
5
6 log_freq_min, log_freq_max = -10, 10
7 x = np.linspace(log_freq_min, log_freq_max, 100)
8 x = 10**x
9
10 plt.plot(x, inst.Omegah2(x), label = "Total")
11 plt.plot(x, inst.Omegah2coll(x), label = "Collision")
12 plt.plot(x, inst.Omegah2sw(x), label = "Sound waves")
13 plt.plot(x, inst.Omegah2turb(x), label = "Turbulence")
14 plt.legend()
15 plt.plot()

```

Listing 10: Computation of the stochastic gravitational waves background spectra.

In Listing 10 a constructor call is used to create an instance `inst` of the class `GWparams.GW_SuperCooled`. The constructor requires as mandatory arguments the percolation temperature T_p (`T_perc`), the efficiency factors α , α_∞ and α_{eq} (`alpha`, `alpha_inf` and `alpha_eq`, respectively), the mean bubble separation at percolation R_{sep*} (`R_star`), the factor γ_* (`gamma_star`), and the Hubble parameter at percolation $H(T_p)$ (`H_star`). The constructor also accepts optional arguments for the sound speed in the plasma at percolation $c_{s,t}(T_p)$ (`c_s`, default `1/np.sqrt(3)`), the bubbles walls speed (`v_w`, default 1), the units of the dimensionful quantities (`units`, default `'GeV'`) and the relativistic degrees of freedom additional to the SM ones (`dark_dof`, default 4).

Once the instance `inst` of `GWparams.GW_SuperCooled` has been constructed, the value of the SGWB density today $h^2\Omega_{GW}(f)$ for each frequency f (in Hertz) can be accessed by simply calling the method `GWparams.GW_SuperCooled.Omegah2`, e.g. `inst.Omegah2(1e-9)` to obtain the SGWB at $f = 10^{-9}$ Hz. In addition, the contributions from each individual GW source can be accessed as well with the methods `GWparams.GW_SuperCooled.Omegah2coll`, `GWparams.GW_SuperCooled.Omegah2sw` and `GWparams.GW_SuperCooled.Omegah2turb`, for the SGWB sourced by bubbles walls collisions, sound waves and turbulence, respectively. In the Listing 10 the class `GWparams.GW_SuperCooled` is used to plot the SGWB spectrum, but the user can take advantage of the modular structure of ELENA to implement this computation in their own software or statistical inference tools.

We report in Figure 18 the SGWB spectra for the two benchmark points in Table 2, together with the periodogram resulting from the NANOGrav 15-years Dataset [1, 66, 67]. As expected, the “Slow” transition results in a much stronger SGWB spectrum, peaked at lower frequencies with respect to the “Fast” one. Only the former can provide a viable solution to the NANOGrav data, with the latter producing a SGWB spectrum which is orders of magnitude smaller in amplitude at the relevant frequencies.

We stress that this is only an example application of the code, and that ELENA can compute the

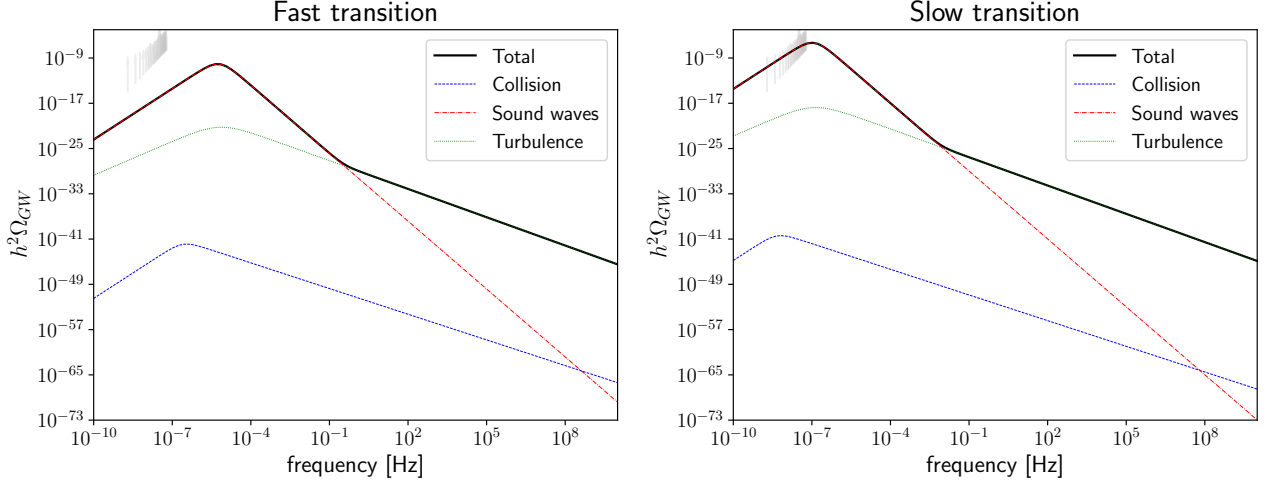


Figure 18: SGWB spectra predicted for the “Fast” (left panel) and “Slow” (right panel) benchmark points in Table 2. Dashed, dot-dashed and dotted lines represent the individual contributions from bubbles walls collisions, sound waves and turbulence, while the continuous black lines are their sum. The gray bands represent the periodogram from the NANOGrav 15-years Dataset [1, 66, 67].

FOPT parameters for general scenarios, not necessarily related to Pulsar Timing Array observatories.

4 Use case: MCMC analysis of a model using PTArcade

In this Section, we show an explicit use case that takes advantage of the speed and modularity of ELENA to perform a statistical analysis on the parameters of a new physics model from observational data. We interface ELENA with PTArcade [68] to perform a Markov Chain Monte Carlo (MCMC) analysis of the dark sector model from Equation (10), introduced in Ref. [31], comparing its predictions for the SGWB to the NANOGrav 15-year data. Notice that the model in Equation (10) needs to be extended to also include portals to the SM sector, to ensure both thermalisation of the dark sector and the decay of its states into SM particles after the phase transition, guaranteeing compatibility with cosmological constraints [13]. In the present analysis, given the scope of the work on the software package ELENA, we focus for simplicity on the minimal dark sector in Equation (10), assuming that the additional interactions with the SM do not significantly affect the FOPT dynamics. We defer the analysis of extensions of Equation (10) to future work, while stressing that the complications from considering a complete model are confined to the particle and cosmology phenomenology sides (e.g. computing the full thermal potential, the dark sector decay widths, or applying the experimental and observational constraints). The pipeline for computing the FOPT thermal parameters with ELENA is unchanged; the user just needs to provide the correct model class for the specific thermal potential. Since a detailed discussion on possible extensions of Equation (10) goes beyond the scope of the present work, we only apply a lower bound on the final reheating temperature, $T_{\text{reh}} \geq 4 \text{ MeV}$ [69], where we estimate [55]

$$T_{\text{reh}} = (1 + \alpha)^{\frac{1}{4}} T_p. \quad (43)$$

Parameter	MAP	Bayes	MLE
$\log_{10} \lambda_\phi$	-2.35	-2.49 ± 0.28	-2.82
$\log_{10} g_D$	-0.16	-0.19 ± 0.07	-0.28
$\log_{10} v_\phi^0 / \text{MeV}$	1.64	2.06 ± 0.57	1.67

Table 3: Maximum a posterior (MAP), Bayes and maximum likelihood (MLE) estimators for the Lagrangian parameters in Equation (10) from the NANOGrav 15-year dataset, obtained from the MCMC sampling performed with PTArcade, as described in the main text.

We use PTArcade to collect more than 9 millions samples⁹ (9,360,965) on the model in Equation (10): this is done by creating a script that contains the code developed in Section 3 and using the final instance of the class `GW_SuperCooled` as input for the `spectrum` function required by PTArcade. For the parameter points where the variable `is_physical` results to `False` we assume no production of SGWB (i.e. $h^2\Omega_{\text{GW}} = 0$ for all frequencies). In Ref. [31] it was shown that, in order to produce a strong FOPT, the full scalar potential generated from the Lagrangian in Equation (10) must have a conformal-like shape, which is realised if the g_D coupling lies close to the line defined by

$$g_D^{\text{roll}} = \left\{ \frac{16\pi^2\lambda_\phi}{3} \left[1 - \frac{\lambda_\phi}{8\pi^2} (5 + 2\log 2) \right] \right\}^{1/4}. \quad (44)$$

To speed-up the convergence of the algorithm, instead of sampling with a uniform prior over the plane $(\log_{10} \lambda_\phi, \log_{10} g_D)$, we assume a uniform probability distribution over the line g_D^{roll} , and a gaussian distribution (centred at g_D^{roll}) over the perpendicular direction. This allows to reduce the burn-in phase, by aligning the sampling directions with the high-likelihood region. In terms of the Lagrangian parameters, the sampling covered the range of values: $\log_{10} \lambda_\phi \in (-3.17, -0.52)$, $\log_{10} g_D \in (-0.67, 0.62)$, $\log_{10} v_\phi^0 / \text{MeV} \in (1.0, 3.4)$ (with a uniform prior distribution for the latter). The results of the analysis, rotated in the physical parameters, are reported in Figure 19, where it is evident that the data strongly prefer a conformal-like potential (i.e. having λ_ϕ and g_D lying around g_D^{roll}) with a new-physics scale below 100 MeV. The maximum a posterior (MAP), Bayes and maximum likelihood (MLE) estimators for the three parameters are identified by a yellow star, a purple rhombus and a red cross, respectively. The MAP, MLE and Bayes estimators for the Lagrangian parameters are reported in Table 3.

To give a more complete idea of the sampled parameter space, we report in a scatter plot all the sampling points generated during the MCMC in Figure 20, with each point coloured according to the log-likelihood value computed by PTArcade. This shows that the employed parametrisation efficiently sampled the phenomenologically relevant region, focusing on the regions with high likelihood but also covering the surrounding areas with lower likelihood values.

We plot the predicted SGWB spectra for the MAP, MLE and Bayes estimator parameters, together with the NANOGrav 15-years periodogram in Figure 21; for completeness we also include the SGWB spectrum from the “Slow” benchmark point in Table 2. A summary of all the discussed points and their thermal parameters as computed by ELENA is reported in Table 4.

⁹The final sample results from the union of 1340 individual chains.

PTArcade MCMC results

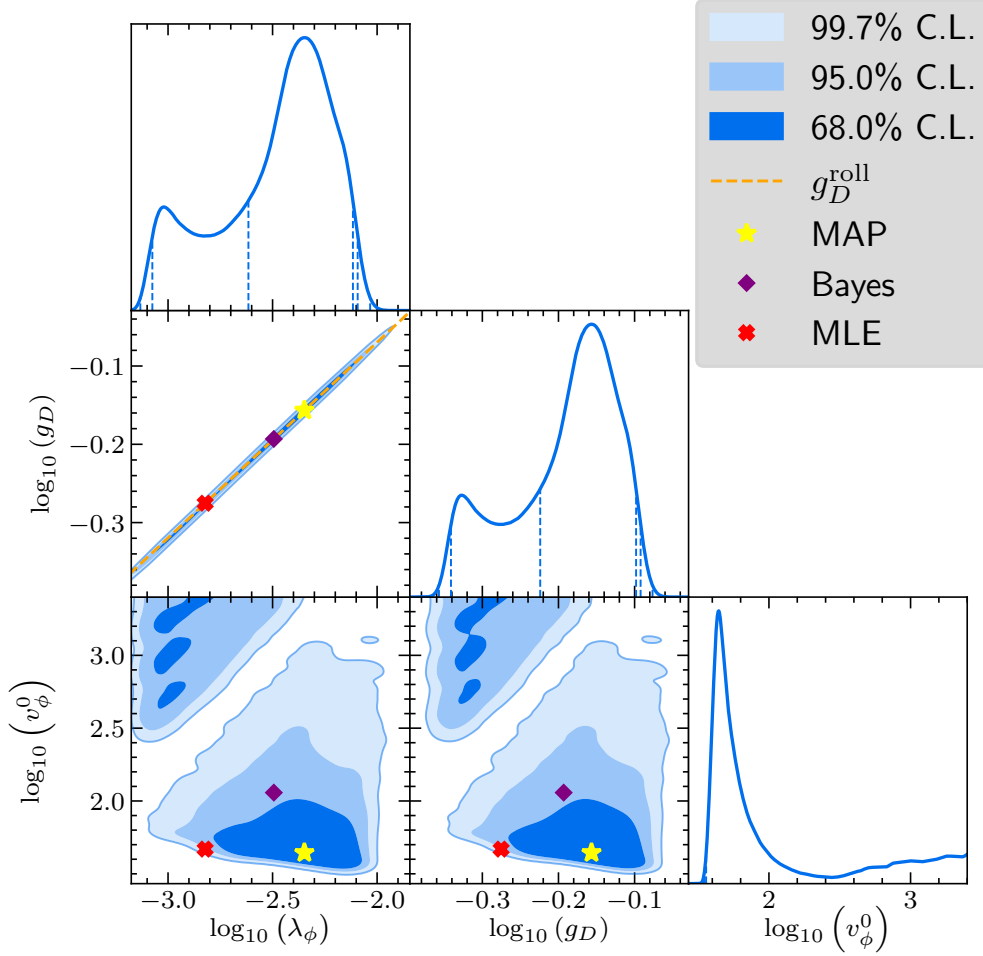


Figure 19: Posteriors for the model parameters in Equation (10) derived from a MCMC study on the NANOGrav 15-year data, performed by interfacing ELENA with PTArcade. We report the regions corresponding to 68% (bright blue), 95% (sky blue) and 99.7% (pale blue) confidence levels, together with the maximum a posteriori (MAP), maximum likelihood (MLE) and Bayes estimators of the three parameters as a yellow star, red cross and purple rhombus, respectively. For reference, we also plot g_D^{roll} in Equation (44) as a dashed orange line.

5 Performance and comparison with CosmoTransitions

To conclude, we compare ELENA with the widely used software CosmoTransitions [19], which employs the commonly used bounce formalism (cf. Equation (1)) to compute the tunnelling action. We compare both the software routines that compute the Euclidean action, as well as the ones that identify the phases of the theory, a step that is required to identify the range of temperatures where a FOPT can happen. All the tests have been performed on a personal machine with an Apple M2 processor.

Concerning the identification of the phases of the theory, we find good agreement among the software, in both the results and the required time, as reported in Table 5.

Some differences are observed in the MLE and “Fast” points, where ELENA takes more time to identify the phases, and quite significantly in the Bayes estimator point, where CosmoTransitions is not able to

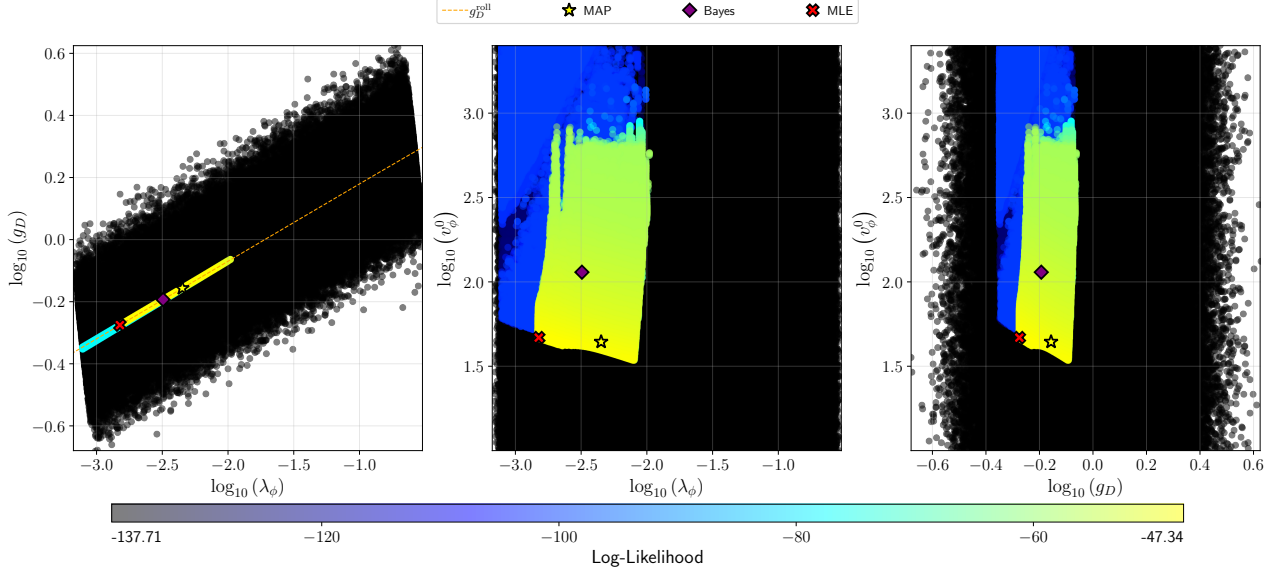


Figure 20: Scatter plots showing the sampled parameter space in two-dimensional slices. The colour bar corresponds to the value of the log-likelihood for each point. Yellow regions provide a larger probability of observing the NANOGrav 15-year signal given the chosen parameters. Black points correspond to parameter values with either no viable FOPT, or for which the variable `is_physical` returns `False`. We highlight the MAP, MLE and Bayes estimators with a yellow star, a red cross and a purple rhombus, respectively. The dashed orange line is g_D^{roll} , Equation (44).

identify the second phase. We verified graphically that at the critical temperature found by ELENA two degenerate local minima exist (cf. Figure 22) and that the one located away from the origin becomes the absolute minimum at lower temperatures, with a barrier persisting until T_{min} , thus confirming the existence of two phases as correctly identified by ELENA.

The results for the computation of the action agree as well for most of the parameter space, as demonstrated in Figure 23 for the “Slow” point in Table 2. For completeness, we also compare in this figure the results of the numerical computations performed by ELENA and CosmoTransitions with the analytic result obtained by fitting the potential to a cubic shape with temperature-dependent polynomial coefficients

$$V(\phi) = m(T)\phi^2 + \eta(T)\phi^3 + \lambda(T)\phi^4. \quad (45)$$

This approximation is sometimes employed in the literature because, for a potential described by Equation (45), the action can be obtained analytically [70]. However, it should be noted that Equation (45) is not guaranteed to provide a good description of the thermal potential in general, as it will be evident in the following.

Overall, the two software agree very well, with a relative difference in the results of order 1% or less for most of the range of temperatures, cf. Figure 24, with the main discrepancies observed at the extreme ends of the temperature range. For $T \sim T_c$ the difference in the action is of order unity, but this has little practical effect since the nucleation rate is infinitesimal at these temperatures. On the other end, for $T \sim T_{\text{min}}$, we notice that CosmoTransitions returns negative values of the action, while the results from ELENA are always positive, and the shape of the action for very low temperatures

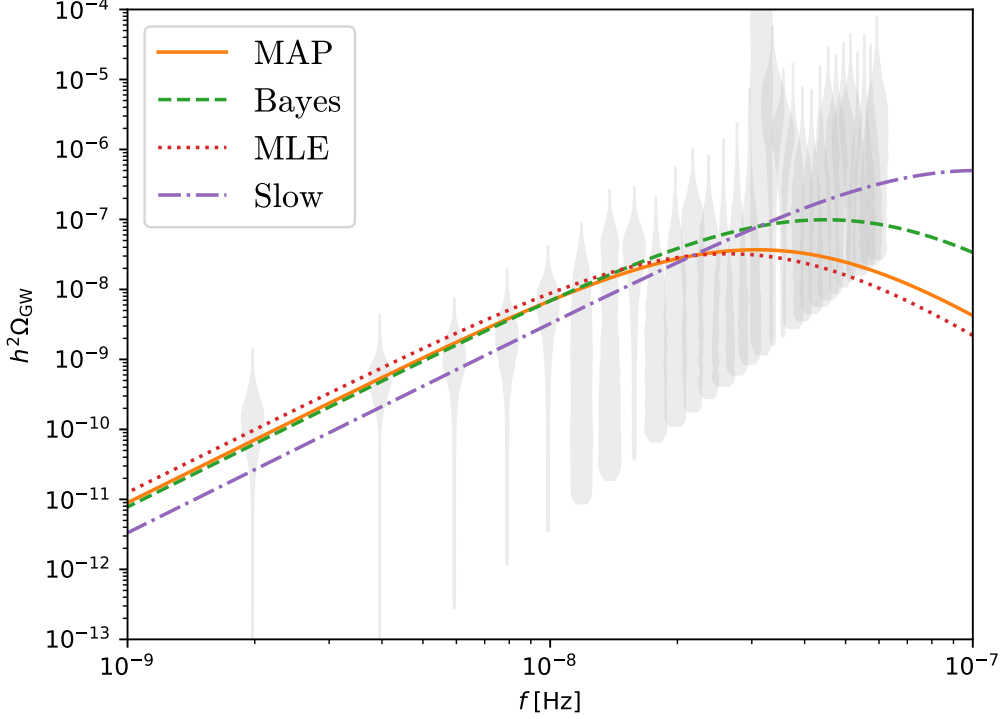


Figure 21: Predicted SGWB spectra for the MAP, MLE and Bayes (central value) estimators in Table 3, compared to the NANOGrav 15-year periodogram. For completeness, we also show the SGWB of the “Slow” point in Table 2 (the signal from the “Fast” point is outside of the plotted region).

does not show any discontinuity, cf. Figure 23. For completeness, we notice that the estimation of the action obtained with the cubic approximation only works in the high-temperature limit ($T \gtrsim 80$ MeV), where the finite-temperature potential can be correctly described by the polynomial expression in Equation (45). For lower temperature values, the conformal shape of the potential becomes relevant, and a simple polynomial expression does not provide a valid fit to it.

To conclude, we comment on the ELENA internal parameters that determine the trade-off between numerical accuracy and speed of computation in the evaluation of the action. These are mainly determined by the optional arguments `step_phi` and `precision` of the class `Vt_vec`, as exemplified in the Listing 3. We show the trade-off between error and time in Figure 25, taking as example the “Slow” point and evaluating the action at half the critical temperature, $T = 57.875$ MeV. We assume as reference (i.e. most precise) value of the action the one obtained with the smallest choice of both parameters, that is `step_phi` = 1e-5 and `precision` = 1e-6. The change in the running time corresponds to the red shaded region for different precisions, to be read on the right vertical axis, while the relative error corresponds to the blue shaded region, to be read on the left vertical axis.

We notice that the dominant parameter in determining the trade-off is `step_phi`, which corresponds to the size of the step in the field dimension, expressed in units of the vev value. This is expected, since this parameter determines how well the potential is sampled over its range, but at the same time, the number of action computations increases with the number of samplings. On the other hand, we find that the relative error does not depend on the `precision` parameter for this point, meaning that the first iteration is already enough to converge to the correct ϕ_0 value; we nevertheless advise against

Quantity	MAP	Bayes	MLE	Slow	Fast
λ_ϕ	4.49×10^{-3}	3.21×10^{-3}	1.51×10^{-3}	6×10^{-3}	1.65×10^{-3}
g_D	0.70	0.64	0.53	0.75	0.54
v_ϕ^0 (MeV)	44.10	1.14×10^2	46.72	5×10^2	5×10^2
m_ϕ (MeV)	4.18	9.14	2.56	54.77	28.72
$m_{Z'}$ (MeV)	30.75	73.16	24.79	3.75×10^2	2.7×10^2
T_{critical} (MeV)	9.50	22.65	7.69	1.16×10^2	84.82
$T_{\text{nucleation}}$ (MeV)	0.91	0.90	0.12	13.94	15.27
$T_{\text{percolation}}$ (MeV)	0.74	0.68	0.11	9.61	14.87
$T_{\text{completion}}$ (MeV)	0.68	0.64	0.10	6.39	14.81
T_{minimal} (MeV)	0.00	0.00	4.13×10^{-2}	0.00	11.55
$T_{\text{reheating}}$ (MeV)	4.46	10.58	4.03	53.84	39.86
P_f^{min}	0.00	2.05×10^{-300}	0.00	6.57×10^{-17}	0.00
α	1.31×10^3	5.71×10^4	1.98×10^6	9.86×10^2	50.59
α_∞	39.79	2.62×10^2	1.64×10^3	34.45	7.62
α_{eq}	11.07	23.94	47.81	11.79	2.92
γ_*	4.26×10^{18}	1.19×10^{18}	6.68×10^{17}	1.69×10^{18}	2.29×10^{16}
γ_{eq}	1.15×10^2	2.38×10^3	4.14×10^4	80.67	14.72
$R_{\text{sep}*}H_*$	0.14	0.23	0.13	0.53	7.13×10^{-3}

Table 4: Numerical values of the model and FOPT thermal parameters for the specific realisations of the minimal dark sector in Equation (10): maximum a posterior (MAP), Bayes and maximum likelihood (MLE) estimators from the MCMC sampling discussed in the main text, plus the “Slow” and “Fast” benchmark points introduced in Table 2.

setting too large values for `precision`, since different potential shapes may require multiple iterations to converge. On general grounds, we find that the default settings of `step_phi` = 1e-3 and `precision` = 1e-3 should provide a valid configuration for most of the users, with a computing time of order 10^{-2} seconds and a result that does not differ more than $\mathcal{O}(10^{-3})$ from the most extreme settings.

6 Conclusion

We introduced **ELENA**: *EvaLuator of tunnelliNg Actions*, a Python package designed to study first-order phase transitions generated by scalar potentials in the Early Universe. The core of the package is a vectorised implementation of the tunnelling potential formalism, which is an alternative method to compute the tunnelling action with respect to the widely adopted bounce equation formalism.

Point	MAP		Bayes		MLE		Slow		Fast	
Software	E	C	E	C	E	C	E	C	E	C
T_{crit} (MeV)	9.50	9.51	22.65	/	7.69	7.69	115.75	115.78	84.82	84.83
T_{min} (MeV)	0	0	0	/	0.04	0	0	0	11.55	11.48
time (s)	0.13	0.14	0.13	0.14	0.39	0.13	0.12	0.13	0.38	0.17

Table 5: Comparison between the values of the critical (T_{crit}) and minimal (T_{min}) temperatures as computed by ELENA (E) and CosmoTransitions (C) for the different benchmark points, as well as of the time taken to identify them. For the Bayes estimator, CosmoTransitions does not identify a second phase; the thermal potential for this point at the critical temperatures identified by ELENA is plotted in Figure 22.

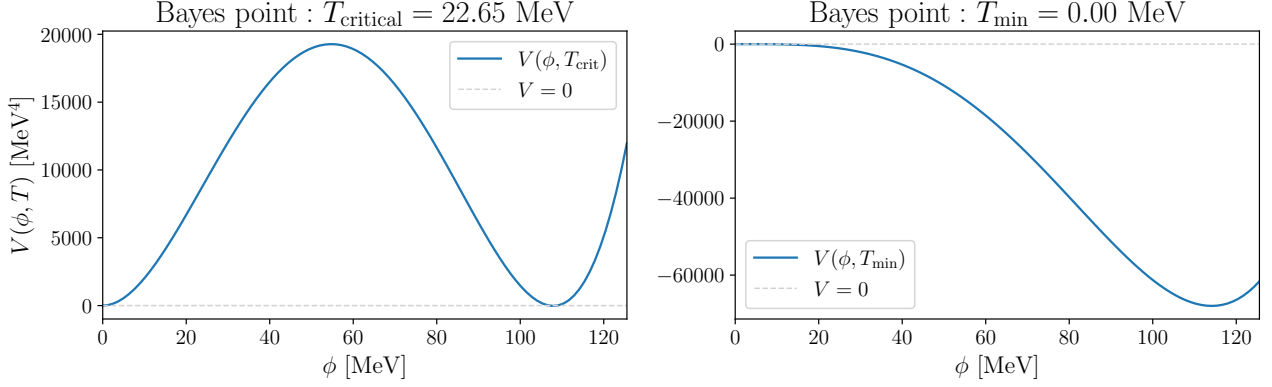


Figure 22: Shape of the thermal potential for the Bayes estimator point at the critical temperature T_{critical} (left) and at the minimal one T_{min} (right).

Contrary to solving the bounce equations of motion, whose solution is a saddle point, in the tunnelling potential approach, the physical action is an absolute minimum in the space of solutions, making the algorithm numerically faster and stable under deviations from the correct solution. To our knowledge, **ELENA** is the first public package that implements this algorithm. We tested **ELENA** against the widely used public software **CosmoTransitions**, finding excellent agreement and faster convergence; in addition, **ELENA** shows strong numerical stability even for very flat potentials, a region where **CosmoTransitions** can deliver unphysical results due to numerical instabilities of the bounce equation solver.

In addition to providing a novel and efficient algorithm to compute the tunnelling action, we ship **ELENA** with a full suite of additional classes and functions to compute the stochastic gravitational waves background generated in a FOPT. We go beyond current codes by avoiding approximations commonly employed in the literature, for instance, not relying on the bag model, using the percolation temperature instead of the nucleation one, explicitly checking the evolution of the physical volume in false vacuum, and computing the mean bubble separation instead of the linear β parameter.

We presented a possible use case for **ELENA**, by interfacing it with the **PTArcade** software to perform a fit of a minimal dark sector to the NANOGrav 15-year data, being able to sample more than 9 millions points in the underlying MCMC study. We nevertheless stress that the scope of **ELENA** is not limited to Pulsar Timing Array Experiments, but can cover the full range of frequencies explored by current and future experiments testing SGWB from FOPT.

Being a first release, **ELENA** also presents some limitation; most notably, it can only deal with single phase transitions (when only two phases at most are simultaneously present) and only accepts single-field potentials. While we plan to generalise the scope of the software in future releases, we point out to users interested in more complex phase histories that the modularity of **ELENA** (where all individual functions and classes are easily accessible) allows for an easy interface with more specialised software [23, 28]. We also notice that the tunnelling potential formalism can be readily generalised to multi-field potentials [25], and we observe no roadblock into implementing this more general algorithm into our numerical routines, other than the work needed to refactor and extend the code. The extension to multi-field potentials will be the focus of the next development phase.

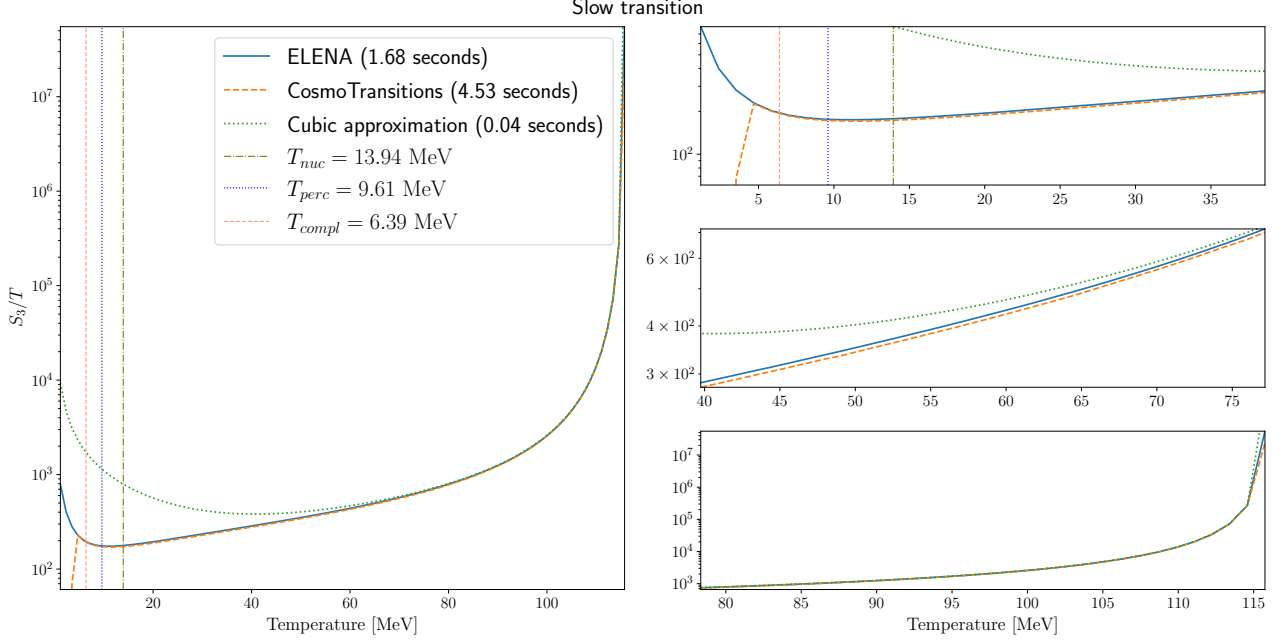


Figure 23: Comparison of the finite-temperature action computation performed by **ELENA**, **CosmoTransitions** and by assuming a cubic potential shape (Equation (45)), for the “Slow” point in Table 2. The left panel shows the full range of temperatures between T_c and T_{min} , while the right subpanels are zoomed over the low (top), intermediate (middle) and high (bottom) temperature values. The time employed by each algorithm to obtain these results is reported in the legend. For reference, vertical lines represent the nucleation (T_n), percolation (T_p) and completion (T_e) temperatures as dashed, dotted and dot-dashed lines, respectively.

Acknowledgements

The numerical results presented in Section 4 have been obtained by employing the Open Physics Hub HPC facility, hosted at the Department of Physics and Astronomy, University of Bologna. ML is funded by the European Union under the Horizon Europe’s Marie Skłodowska-Curie project 101068791 — NuBridge. F. C. acknowledges support from the FORTE project CZ.02.01.01/00/22 008/0004632 co-funded by the EU and the Ministry of Education, Youth and Sports of the Czech Republic. JHZ is supported by the National Science Centre, Poland (research grant No. 2021/42/E/ST2/00031). In the initial stages of this project, the research of FC, JHZ, SP and SRA received support from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860881-HIDDeN. ML thanks Fermilab for hosting him during the development of part of this work.

A Installation example using the **conda** package manager

ELENA is designed to be as much a self-contained Python package as possible, only requiring a small number of dependencies. Specifically, it depends on **CosmoTransitions** for constructing the finite-temperature class model, plus the **NumPy** and **SciPy** libraries to perform numerical operations. To fully run the notebook examples shipped with the software some additional dependencies are required,

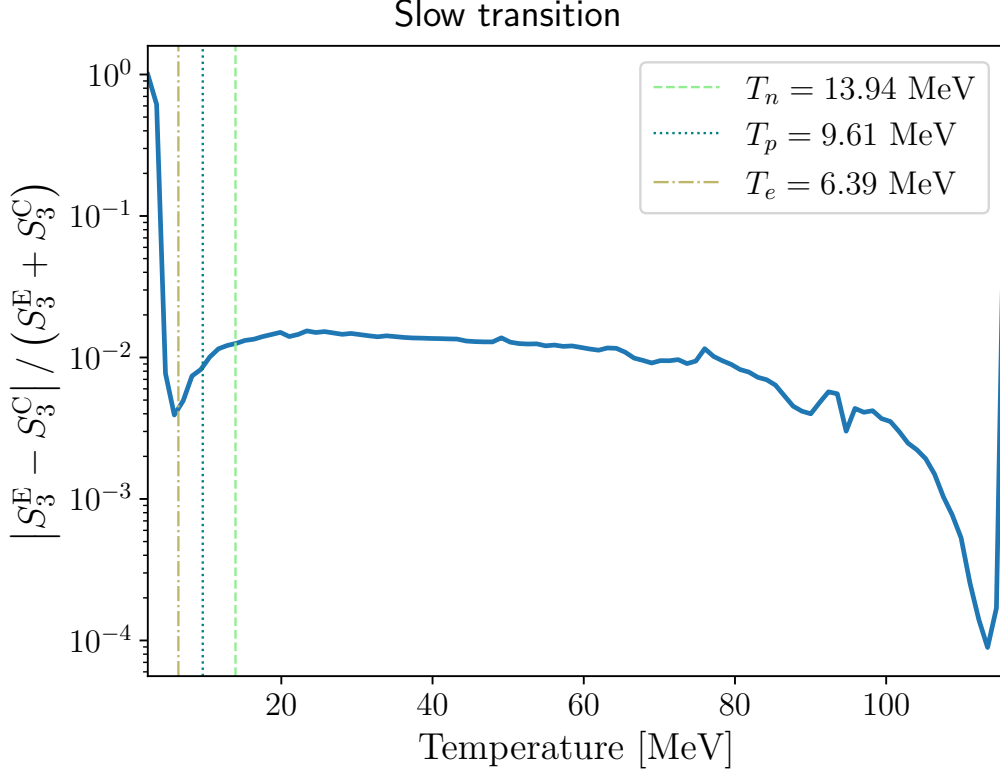


Figure 24: Relative difference of the finite-temperature tunnelling action computed by ELENA (S_3^E) and CosmoTransitions (S_3^C), as a function of temperature, for the “Slow” point in Table 2. Vertical lines represent the nucleation (T_n), percolation (T_p) and completion (T_e) temperatures as dashed, dotted and dot-dashed lines, respectively.

specifically for data visualisation and interactive sessions, but we stress that these are not required to run ELENA itself.

In the following, we show how to use the `conda` package manager to create a self-contained environment where ELENA can be executed. The user simply needs to run the following commands in the terminal:

```
conda create --name elena -c conda-forge python numpy scipy cosmoTransitions matplotlib
ipykernel natpy la_forge
conda activate elena
```

At this point, a `conda` environment containing the required dependencies will be created and activated. The next step is to download the ELENA code:

```
git clone https://github.com/michelelucente/ELENA
cd ELENA
```

ELENA is now present on the local machine, and the user can test it by running for instance the Jupyter notebook located in `'./examples/phase_transition.ipynb'`.

References

- [1] NANOGrav collaboration, *The NANOGrav 15 yr Data Set: Evidence for a Gravitational-wave*

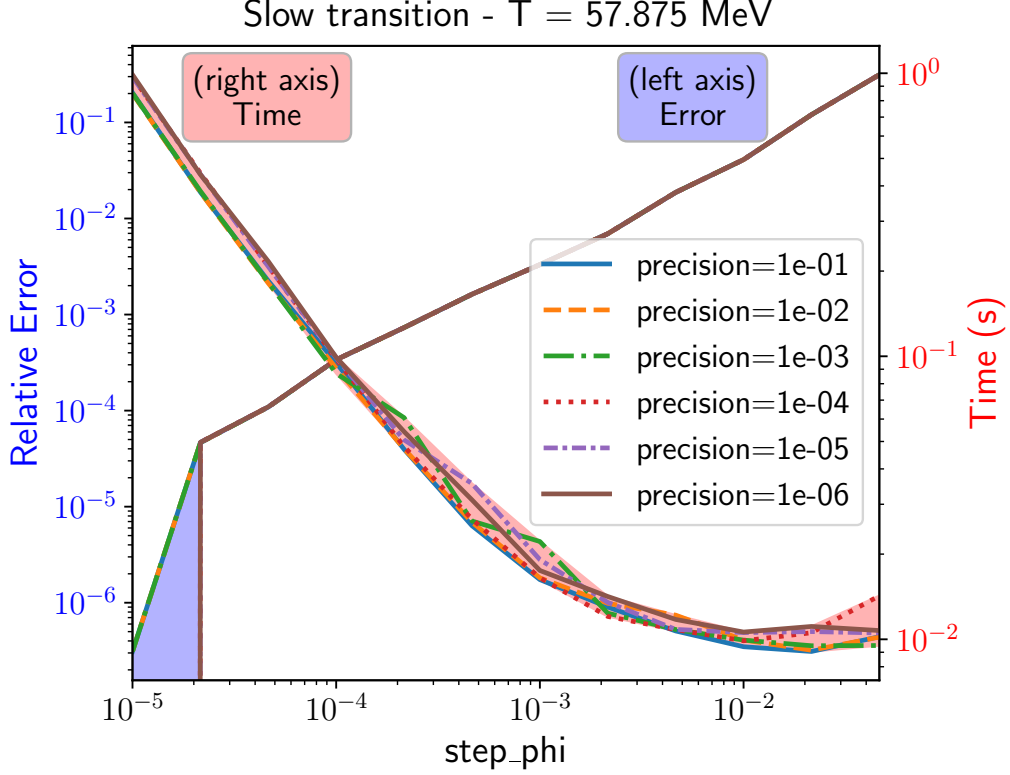


Figure 25: Trade-off between accuracy and computational time in ELENA for the computation of the action, using the “Slow” point in Table 2 at $T = T_c/2$ as benchmark scenario. The internal parameters determining the trade-off are `step_phi` (varied over the horizontal axis) and `precision` (with different values represented by the different line-styles) in the class `Vt_vec`. For each combination of parameters, the relative error of the computation is plotted with a light blue background (values to be read on the left vertical axis), while the required computational time is plotted with a light red background (values to be read on the right vertical axis).

Background, Astrophys. J. Lett. **951** (2023) L8 [2306.16213].

- [2] EPTA, INPTA: collaboration, *The second data release from the European Pulsar Timing Array - III. Search for gravitational wave signals*, *Astron. Astrophys.* **678** (2023) A50 [2306.16214].
- [3] D.J. Reardon et al., *Search for an Isotropic Gravitational-wave Background with the Parkes Pulsar Timing Array*, *Astrophys. J. Lett.* **951** (2023) L6 [2306.16215].
- [4] H. Xu et al., *Searching for the Nano-Hertz Stochastic Gravitational Wave Background with the Chinese Pulsar Timing Array Data Release I*, *Res. Astron. Astrophys.* **23** (2023) 075024 [2306.16216].
- [5] M. Rajagopal and R.W. Romani, *Ultra-Low-Frequency Gravitational Radiation from Massive Black Hole Binaries*, *Astrophys. J.* **446** (1995) 543 [astro-ph/9412038].
- [6] J.S.B. Wyithe and A. Loeb, *Low-frequency gravitational waves from massive black hole binaries: Predictions for lisa and pulsar timing arrays*, *The Astrophysical Journal* **590** (2003) 691.

- [7] A.H. Jaffe and D.C. Backer, *Gravitational waves probe the coalescence rate of massive black hole binaries*, *The Astrophysical Journal* **583** (2003) 616.
- [8] Y. Gouttenoire, *First-Order Phase Transition Interpretation of Pulsar Timing Array Signal Is Consistent with Solar-Mass Black Holes*, *Phys. Rev. Lett.* **131** (2023) 171404 [2307.04239].
- [9] Z.-C. Chen, S.-L. Li, P. Wu and H. Yu, *NANOGrav hints for first-order confinement-deconfinement phase transition in different QCD-matter scenarios*, *Phys. Rev. D* **109** (2024) 043022 [2312.01824].
- [10] D. Wang, *Constraining Cosmological Phase Transitions with Chinese Pulsar Timing Array Data Release 1*, 2307.15970.
- [11] T. Ghosh, A. Ghoshal, H.-K. Guo, F. Hajkarim, S.F. King, K. Sinha et al., *Did we hear the sound of the Universe boiling? Analysis using the full fluid velocity profiles and NANOGrav 15-year data*, *JCAP* **05** (2024) 100 [2307.02259].
- [12] A. Addazi, Y.-F. Cai, A. Marciano and L. Visinelli, *Have pulsar timing array methods detected a cosmological phase transition?*, *Phys. Rev. D* **109** (2024) 015028 [2306.17205].
- [13] T. Bringmann, P.F. Depta, T. Konstandin, K. Schmidt-Hoberg and C. Tasillo, *Does NANOGrav observe a dark sector phase transition?*, *JCAP* **11** (2023) 053 [2306.09411].
- [14] D. Croon and D.J. Weir, *Gravitational Waves from Phase Transitions*, 2410.21509.
- [15] M.W. Winkler and K. Freese, *Origin of the Stochastic Gravitational Wave Background: First-Order Phase Transition vs. Black Hole Mergers*, 2401.13729.
- [16] A. Conaci, L. Delle Rose, P.S.B. Dev and A. Ghoshal, *Slaying axion-like particles via gravitational waves and primordial black holes from supercooled phase transition*, *JHEP* **12** (2024) 196 [2401.09411].
- [17] K. Fujikura, S. Girmohanta, Y. Nakai and M. Suzuki, *NANOGrav signal from a dark conformal phase transition*, *Phys. Lett. B* **846** (2023) 138203 [2306.17086].
- [18] A. Banik, Y. Cui, Y.-D. Tsai and Y. Tsai, *The Sound of Dark Sectors in Pulsar Timing Arrays*, 2412.16282.
- [19] C.L. Wainwright, *CosmoTransitions: Computing Cosmological Phase Transition Temperatures and Bubble Profiles with Multiple Fields*, *Comput. Phys. Commun.* **183** (2012) 2006 [1109.4189].
- [20] P. Basler, L. Biermann, M. Mühlleitner, J. Müller, R. Santos and J.a. Viana, *BSMPT v3 A Tool for Phase Transitions and Primordial Gravitational Waves in Extended Higgs Sectors*, 2404.19037.

- [21] F. Ertas, F. Kahlhoefer and C. Tasillo, *Turn up the volume: listening to phase transitions in hot dark sectors*, *JCAP* **02** (2022) 014 [2109.06208].
- [22] V. Brdar, M. Finetti, M. Matteini, A.P. Morais and M. Nemevšek, *PT2GWFinder: A Package for Cosmological First-Order Phase Transitions and Gravitational Waves*, 2505.04744.
- [23] P. Athron, C. Balazs, A. Fowlie, L. Morris, W. Searle, Y. Xiao et al., *PhaseTracer2: from the effective potential to gravitational waves*, *Eur. Phys. J. C* **85** (2025) 559 [2412.04881].
- [24] J.R. Espinosa, *A Fresh Look at the Calculation of Tunneling Actions*, *JCAP* **07** (2018) 036 [1805.03680].
- [25] J.R. Espinosa and T. Konstandin, *A Fresh Look at the Calculation of Tunneling Actions in Multi-Field Potentials*, *JCAP* **01** (2019) 051 [1811.09185].
- [26] S.R. Coleman, *The Fate of the False Vacuum. 1. Semiclassical Theory*, *Phys. Rev. D* **15** (1977) 2929.
- [27] P. Athron, C. Balázs, M. Bardsley, A. Fowlie, D. Harries and G. White, *BubbleProfiler: finding the field profile and action for cosmological phase transitions*, *Comput. Phys. Commun.* **244** (2019) 448 [1901.03714].
- [28] P. Athron, C. Balázs, A. Fowlie and Y. Zhang, *PhaseTracer: tracing cosmological phases and calculating transition properties*, *Eur. Phys. J. C* **80** (2020) 567 [2003.02859].
- [29] V. Guada, M. Nemevšek and M. Pintar, *FindBounce: Package for multi-field bounce actions*, *Comput. Phys. Commun.* **256** (2020) 107480 [2002.00881].
- [30] A. Ekstedt, P. Schicho and T.V.I. Tenkanen, *DRalgo: A package for effective field theory approach for thermal phase transitions*, *Comput. Phys. Commun.* **288** (2023) 108725 [2205.08815].
- [31] F. Costa, J. Hoefken Zink, M. Lucente, S. Pascoli and S. Rosauero-Alcaraz, *Supercooled dark scalar phase transitions explanation of NANOGrav data*, *Phys. Lett. B* **868** (2025) 139634 [2501.15649].
- [32] Y. Zhu, J. Liu, R. Qin and L. Bian, *Theoretical Uncertainties in First-Order Electroweak Phase Transitions*, 2503.19566.
- [33] M. Kierkla, B. Swiezevska, T.V.I. Tenkanen and J. van de Vis, *Gravitational waves from supercooled phase transitions: dimensional transmutation meets dimensional reduction*, *JHEP* **02** (2024) 234 [2312.12413].
- [34] B. Sojka and B. Swiezevska, *Radiative symmetry breaking from the on-shell perspective*, *Phys. Lett. B* **860** (2025) 139155 [2407.07437].

- [35] S.R. Coleman and E.J. Weinberg, *Radiative Corrections as the Origin of Spontaneous Symmetry Breaking*, *Phys. Rev. D* **7** (1973) 1888.
- [36] A.D. Linde, *Fate of the False Vacuum at Finite Temperature: Theory and Applications*, *Phys. Lett. B* **100** (1981) 37.
- [37] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau et al., *Array programming with NumPy*, *Nature* **585** (2020) 357.
- [38] A.N. Kolmogorov, *On the statistical theory of the crystallization of metals*, *Bull Acad Sci URSS (Cl Sci Math Nat)* **3** (1937) 335.
- [39] W.A. Johnson, *Reaction kinetics in process of nucleation and growth*, *Transactions of Transactions of the American Institute of Mining and Metallurgical Engineers* **135** (1939) 416.
- [40] M. Avrami, *Kinetics of phase change. i general theory*, *The Journal of chemical physics* **7** (1939) 1103.
- [41] M. Avrami, *Kinetics of phase change. ii transformation-time relations for random distribution of nuclei*, *The Journal of chemical physics* **8** (1940) 212.
- [42] M. Avrami, *Granulation, phase change, and microstructure kinetics of phase change. iii*, *The Journal of chemical physics* **9** (1941) 177.
- [43] A.H. Guth and S.H.H. Tye, *Phase Transitions and Magnetic Monopole Production in the Very Early Universe*, *Phys. Rev. Lett.* **44** (1980) 631.
- [44] A.H. Guth and E.J. Weinberg, *Cosmological Consequences of a First Order Phase Transition in the $SU(5)$ Grand Unified Model*, *Phys. Rev. D* **23** (1981) 876.
- [45] P. Athron, C. Balázs, A. Fowlie, L. Morris and L. Wu, *Cosmological phase transitions: From perturbative particle physics to gravitational waves*, *Prog. Part. Nucl. Phys.* **135** (2024) 104094 [2305.02357].
- [46] A. Chodos, R.L. Jaffe, K. Johnson, C.B. Thorn and V.F. Weisskopf, *A New Extended Model of Hadrons*, *Phys. Rev. D* **9** (1974) 3471.
- [47] P. Athron, C. Balázs and L. Morris, *Supercool subtleties of cosmological phase transitions*, *JCAP* **03** (2023) 006 [2212.07559].
- [48] S.R. Broadbent and J.M. Hammersley, *Percolation processes: I. Crystals and mazes*, *Proc. Cambridge Phil. Soc.* **53** (1957) 629.
- [49] V.K. Shante and S.K. and, *An introduction to percolation theory*, *Advances in Physics* **20** (1971) 325 [<https://doi.org/10.1080/00018737100101261>].
- [50] A. Hunt, R. Ewing and B. Ghanbarian, *Percolation Theory for Flow in Porous Media*, Lecture Notes in Physics, Springer International Publishing (2014).

- [51] C.D. Lorenz and R.M. Ziff, *Precise determination of the critical percolation threshold for the three-dimensional “swiss cheese” model using a growth algorithm*, *The Journal of Chemical Physics* **114** (2001) 3659
[https://pubs.aip.org/aip/jcp/article-pdf/114/8/3659/19017114/3659_1_online.pdf].
- [52] J. Lin and H. Chen, *Continuum percolation of porous media via random packing of overlapping cube-like particles*, *Theoretical and Applied Mechanics Letters* **8** (2018) 299.
- [53] M. Li, H. Chen and J. Lin, *Numerical study for the percolation threshold and transport properties of porous composites comprising non-centrosymmetrical superovoidal pores*, *Computer Methods in Applied Mechanics and Engineering* **361** (2020) 112815.
- [54] M.S. Turner, E.J. Weinberg and L.M. Widrow, *Bubble nucleation in first order inflation and other cosmological phase transitions*, *Phys. Rev. D* **46** (1992) 2384.
- [55] J. Ellis, M. Lewicki and J.M. No, *On the Maximal Strength of a First-Order Electroweak Phase Transition and its Gravitational Wave Signal*, *JCAP* **04** (2019) 003 [1809.08242].
- [56] R. Jinno and M. Takimoto, *Gravitational waves from bubble collisions: An analytic derivation*, *Phys. Rev. D* **95** (2017) 024009 [1605.01403].
- [57] M. Lewicki and V. Vaskonen, *Gravitational waves from bubble collisions and fluid motion in strongly supercooled phase transitions*, *Eur. Phys. J. C* **83** (2023) 109 [2208.11697].
- [58] J.R. Espinosa, T. Konstandin, J.M. No and G. Servant, *Energy Budget of Cosmological First-order Phase Transitions*, *JCAP* **06** (2010) 028 [1004.4187].
- [59] J. Ellis, M. Lewicki, J.M. No and V. Vaskonen, *Gravitational wave energy budget in strongly supercooled phase transitions*, *JCAP* **06** (2019) 024 [1903.09642].
- [60] F. Giese, T. Konstandin and J. van de Vis, *Model-independent energy budget of cosmological first-order phase transitions—A sound argument to go beyond the bag model*, *JCAP* **07** (2020) 057 [2004.06995].
- [61] F. Giese, T. Konstandin, K. Schmitz and J. van de Vis, *Model-independent energy budget for LISA*, *JCAP* **01** (2021) 072 [2010.09744].
- [62] D. Bodeker and G.D. Moore, *Can electroweak bubble walls run away?*, *JCAP* **05** (2009) 009 [0903.4099].
- [63] D. Bodeker and G.D. Moore, *Electroweak Bubble Wall Speed Limit*, *JCAP* **05** (2017) 025 [1703.08215].
- [64] Y. Gouttenoire, R. Jinno and F. Sala, *Friction pressure on relativistic bubble walls*, *JHEP* **05** (2022) 004 [2112.07686].

- [65] M. Hindmarsh, S.J. Huber, K. Rummukainen and D.J. Weir, *Numerical simulations of acoustically generated gravitational waves at a first order phase transition*, *Phys. Rev. D* **92** (2015) 123009 [[1504.03291](#)].
- [66] NANOGrav collaboration, *The NANOGrav 15 yr Data Set: Observations and Timing of 68 Millisecond Pulsars*, *Astrophys. J. Lett.* **951** (2023) L9 [[2306.16217](#)].
- [67] T.N. Collaboration, *Kde representations of the gravitational wave background free spectra present in the nanograv 15-year dataset*, June, 2023. [10.5281/zenodo.10344086](#).
- [68] A. Mitridate, D. Wright, R. von Eckardstein, T. Schröder, J. Nay, K. Olum et al., *PTArcade*, [2306.16377](#).
- [69] S. Hannestad, *What is the lowest possible reheating temperature?*, *Phys. Rev. D* **70** (2004) 043506 [[astro-ph/0403291](#)].
- [70] M. Matteini, M. Nemevšek, Y. Shoji and L. Ubaldi, *False vacuum decay rate from thin to thick walls*, *JHEP* **04** (2025) 120 [[2404.17632](#)].