Basic documentation for lattice simulations

Lauri Niemi

August 26, 2019

1 Multicanonical algorithm

In a first-order phase transition, mixed-phase configurations are exponentially suppressed by a factor proportional to the volume, $p \sim e^{-V}$. Thus, for large lattices it becomes necessary to enhance tunneling probability in order to study phase transitions. For this we use the multicanonical method [1]: our action is modified by a weight function $W(R)$, where R is a suitable order parameter that distinguishes all relevant phases. We choose sign convention $S' = S_{\text{canonical}} + W(R)$, so that weight *increases* when a certain configuration is to be suppressed and the ideal weight function is just the canonical probability distribution for R. Once we obtain a multicanonical distribution from a simulation, the canonical distribution is obtained simply by reweighting. **In the measurement file, we measure $-W(R)$ instead of $W(R)$ to ensure compability with Kari.**

In practice the weight function is implemented by specifying a range $R_{\min} \leq R \leq R_{\max}$ that we want to enhance with multicanonical, and dividing this range into bins of equal length so that the system is in bin i when $R_i \leq R < R_{i+1}$, and $R_0 = R_{\min}$. Inside bin i, we approximate W by a linear interpolation. Denoting $W(R_i) = w_i$, for $R_i \leq R < R_{i+1}$ we have

$$W(R) = w_i + (R - R_i)\frac{w_{i+1} - w_i}{R_{i+1} - R_i}, \tag{1}$$

and in the last bin we simply use constant weight. Outside the binning range the weight is chosen to be constant and equal to w_i in the nearest bin. For nucleation we would need infinite weighting outside the range. This is also implemented and can be specified in config file.

Since the weight function generally depends on a *global* parameter R (ex. volume average of $\phi^\dagger\phi$), parallelization of multicanonical algorithms is not straightforward. We use the following method, adapted from Kari:

- Perform an even-odd sweep on the lattice, updating half of the sites locally using *canonical* updates such as overrelaxation. This moves the system towards a local minimum of the canonical ensemble. For example if the order parameter is $\phi^\dagger\phi$, we update the Higgs field at half of the sites but do not touch the other fields yet.

- After the sweep, recalculate the order parameter and perform an accept/reject step based on the change in the weight function. The sweep of updates is accepted with probability $\min(1, \exp[W(R) - W(R')])$, where R and R' are the old and new values the order parameter. If rejected, ALL local updates contributing to the weight change are undone (ex. if R is the Higgs hopping term, we sweep and update gauge links and the Higgs, and then apply the multicanonical step. If rejected, undo changes in both Higgs and the links). This step produces a bias towards mixed-phase configurations where the weight function is smaller.

For the "easy" order parameter $\phi^\dagger\phi$, the acceptance rate of the multicanonical update is generally good, $> 90\%$. This is in agreement with Kari's acceptance rates in the SM and MSSM.

To do: optimize the algorithm for more complicated order parameters. It should be safe to do perform the multicanonical step in parts, so that if we choose hopping term as the order parameter, we can first update half of the links, do global acc/rej, then update half of the Higgs and do acc/rej again etc.

1.1 Calculation of the weight function

Since the canonical probability distribution (and the weight) is not known *a priori*, we need an algorithm for approximating the weight function. For our purposes there are (at least) two good options for automatizing this process:

- Recursive computation of Berg. The detailed recipe for performing (a variation of) this algorithm has been described in [2] (small detail: weight in the first bin can be set arbitrary, because weights in all other bins are calculated relative to it). This method is very general and will eventually converge to the correct weight function, but is somewhat complicated and can be hard to implement well. **We will not use this.**

- The Landau-Wang method. We use a variation of this, basically copied from Kari's old lecture notes (he also uses it in his current SUSY code).

Details of our algorithm are as follows:

- Perform a short Monte Carlo run with fixed weight, using the multicanonical update outlined above. After each multicanonical acceptance step (even if rejected!), measure the order parameter and check which bin the system is in. Unless the order parameter is outside the binning range, increase w.hits[bin] by one.

- After M ($\sim 500 - 2000$) such measurements, update weight function in each bin by factor w.hits[bin]$\times\delta/n_{\text{bins}}$, where δ is an increment factor and n_{bins} is the total number of bins (the normalization is very convenient and generally leads to smoother weight functions). After updating, reset w.hits and repeat the measurements.

- Initially when the weight is flat, canonical configurations are preferred and so weight is increased in the pure phases by large amounts. To make the weight function converge, the increment factor δ needs to be gradually decreased so that weight of mixed-phase configurations does not catch up to the pure phases. Again we adapt Kari's method here: each time the system tunnels from the last bin to the first bin (or vice versa), set $\delta \leftarrow \delta/1.5$. This check is only performed after a full set of multicanonical measurements (after updating the weight).

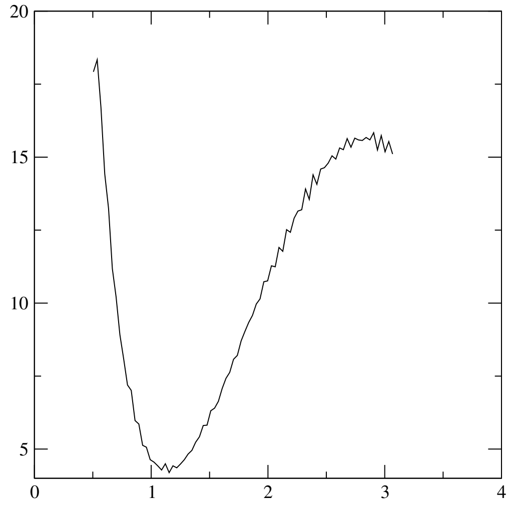

Figure 1: Sample weight function produced using the above algorithm. This corresponds to the SM case with $m_H^* = 35$ GeV, $\beta_G = 8$ and $V = 10\times 10\times 30$. Horizontal axis is the multicanonical order parameter $\langle\frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger\Phi\rangle$.

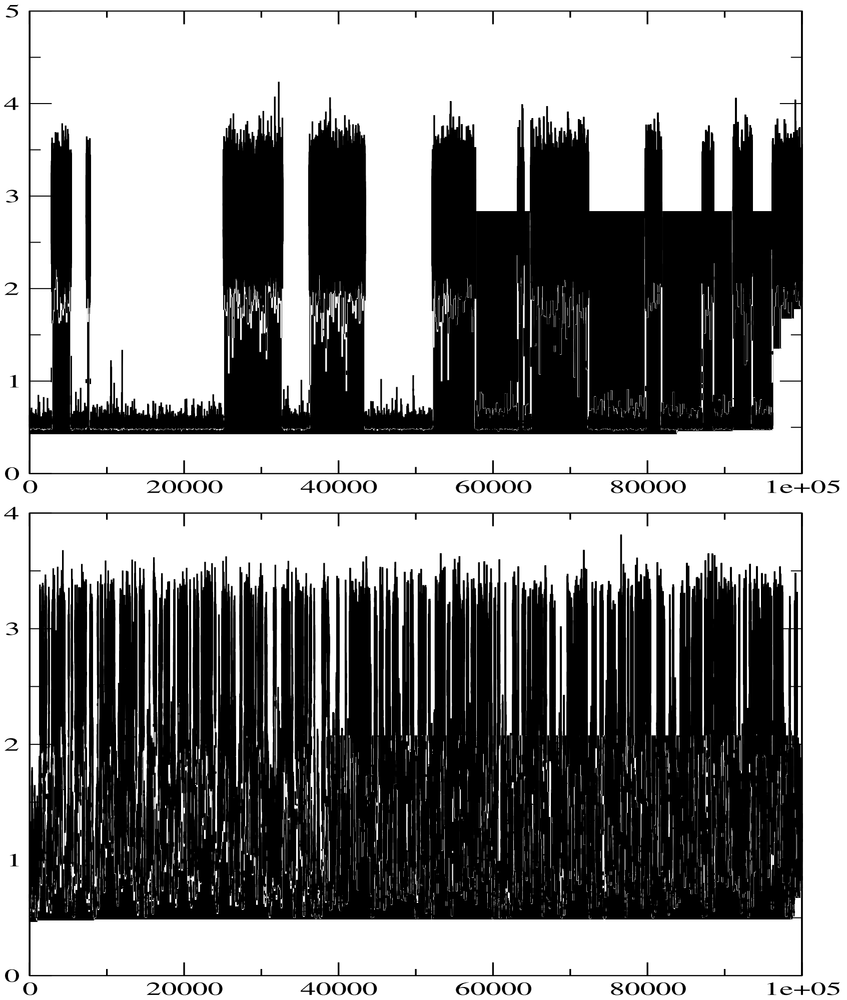

Figure 2: Time histories for the observable $\langle \frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger\Phi\rangle$, top: no multicanonical weight, bottom: using the weight in Fig. 1. Without the weight function, the system is stuck in the pure phases and only rarely samples the mixed-phase configurations. Here the volume is still quite small; already for a slightly larger volume **I could not get the system to tunnel even once without multicanonical weighting.**

A possible downside of this method is that it may not be possible to produce a completely smooth weight function if the δ parameter becomes small too quickly. Usually this does not matter however, because the resulting "spikes" remain quite small. A good initial value for δ is ~ 0.5 and number of bins ~ 100. Binning range should be chosen so that both spikes are included and preferably some extra, but not regions that are already very strongly suppressed (need to be especially careful with the symmetric phase peak that is usually very sharp).

2 Lattice actions and parametrization

By default, the code assumes that all physical input parameters and fields are in natural lattice units, i.e. scaled dimensionless by the lattice spacing a. Here we describe the procedure for converting the continuum parameters into input parameters for the code for some implemented models.

2.1 SU(2) + fundamental Higgs

This is simply the dimensionally reduced Standard Model with U(1) hypercharge neglected. Simulations are to be compared to results of Ref. [3]. Our 3d continuum Lagrangian (with couplings chosen to be dimensionful, so that the action $S = \int d^3x\mathcal{L}$) is

$$\mathcal{L} = \frac{1}{4}(F^a_{ij})^2 + (D_i\phi)^\dagger(D_i\phi) + \mu^2_\phi\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2. \tag{2}$$

Everything here is in terms of 3d parameters, hence no subscript. On the lattice, it is convenient to parametrize the doublet using the Pauli matrices σ_j; this simplifies many update algorithms. We define

$$\Phi = \frac{1}{\sqrt{2}}(\phi_0\sigma_0 + i\phi_k\sigma_k), \tag{3}$$

where σ_0 is the unit matrix and ϕ_μ are real numbers. In terms of the components ϕ_μ, the familiar vector ϕ is

$$\phi = \frac{1}{\sqrt{2}}\begin{pmatrix}\phi_2 - i\phi_1\\ \phi_0 + i\phi_3\end{pmatrix}, \tag{4}$$

which can be checked by comparing the hopping term where the ordering of the components does make a difference. Note that $\frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger\Phi = \phi^\dagger\phi = \frac{1}{2}\phi_\mu\phi_\mu$. In David's code, the components are ordered differently as the vector notation is preferred, leading to a seemingly different implementation of the hopping term.

The discretized action reads (i, j denote directions)

$$\begin{aligned} S_{\text{latt}} =& \beta_G \sum_{\mathbf{x}, i<j} \left(1 - \frac{1}{2}P_{ij}(x)\right) \\ &+ 2\sum_{\mathbf{x},i}\left(a\frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger(x)\Phi(x) - a\frac{1}{2}\,\mathrm{Re}\,\mathrm{Tr}\,\Phi^\dagger(x)U_i(x)\Phi(x+i)\right) \\ &+ \sum_{\mathbf{x}}\left(a^2m^2_\phi\frac{1}{2}a\,\mathrm{Tr}\,\Phi^\dagger\Phi + a\lambda\left[\frac{1}{2}a\,\mathrm{Tr}\,\Phi^\dagger\Phi\right]^2\right), \end{aligned} \tag{5}$$

where U_i is a SU(2) link matrix in direction i, related to the gauge field A^a_i as

$$U_i(x) = \exp\left[\frac{1}{2}ig\sigma_a A^a_i(x)\right], \tag{6}$$

and $P_{ij}(x) = \mathrm{Re}\,\mathrm{Tr}\,U_i(x)U_j(x+i)U^\dagger_i(x+j)U^\dagger_j(x)$. In the code, we write the link as

$$U_i = u_0\sigma_0 + iu_k\sigma_k. \tag{7}$$

Note the different normalization compared to the doublet. The condition $\det U = 1$ is required for U to be an SU(2) matrix; otherwise it is an $U(2)$ matrix. For SU(2), all traces here are automatically real.

The code operates with the dimensionless object $\frac{1}{2}a\,\mathrm{Tr}\,\Phi^\dagger\Phi$ etc, and takes as inputs the combinations $m^2_{\text{lat}} \equiv a^2m^2_\phi$, $\lambda_{\text{lat}} \equiv a\lambda$ and

$$\beta_G = \frac{4}{ag^2}, \tag{8}$$

with β_G essentially used to fix the lattice spacing. The lattice parameters need to be matched onto parameters in continuum renormalization (usually $\overline{\text{MS}}$) by comparing divergences at two loops (c.f. Ref. [7]). Formulas for converting continuum parameters into inputs for the code are listed in Appendix A.

2.2 SU(2) + adjoint scalar (real triplet)

This is essentially a dimensionally reduced QCD with two colors (see Ref. [4]). The adjoint scalar is defined as $\Sigma = \frac{1}{2}\Sigma^a \sigma_a$ (3 components) and the 3d continuum Lagrangian is

$$\mathcal{L} = \frac{1}{4}(F^a_{ij})^2 + \frac{1}{2}(D_i \Sigma^a)^2 + \frac{1}{2}\mu^2_\Sigma \Sigma^a \Sigma^a + \frac{1}{4} b_4 (\Sigma^a \Sigma^a)^2. \tag{9}$$

Again, we prefer to write the lattice action using the matrix Σ, keeping in mind that $\mathrm{Tr}\,\Sigma^2 = \frac{1}{2}\Sigma^a\Sigma^a$. It reads

$$\begin{aligned} S_{\text{latt}} =& \beta_G \sum_{\mathbf{x}, i<j} \Big(1 - \frac{1}{2}P_{ij}(x)\Big) + 2\sum_{\mathbf{x},i} \Big(a\,\mathrm{Tr}\,\Sigma(x)^2 - a\Sigma(x)U_i(x)\Sigma(x+i)U^\dagger_i(x)\Big) \\ &+ \sum_{\mathbf{x}} \Big[(am_\Sigma)^2(a\,\mathrm{Tr}\,\Sigma^2) + ab_4(a\,\mathrm{Tr}\,\Sigma^2)^2\Big], \end{aligned} \tag{10}$$

This is exactly the same action as in Ref. [4], but they denote $\Sigma = A_0, m_\Sigma = m_D, b_4 = \lambda_A$.

2.3 SU(2) + fundamental Higgs + adjoint scalar

The full continuum Lagrangian in 3d is

$$\begin{aligned} \mathcal{L} =& \frac{1}{4}(F^a_{ij})^2 + (D_i\phi)^\dagger(D_i\phi) + \mu^2_\phi \phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2 \\ &+ \frac{1}{2}(D_i\Sigma^a)^2 + \frac{1}{2}\mu^2_\Sigma\Sigma^a\Sigma^a + \frac{1}{4}b_4(\Sigma^a\Sigma^a)^2 + \frac{1}{2}a_2\phi^\dagger\phi\Sigma^a\Sigma^a, \end{aligned} \tag{11}$$

and the lattice action is simply a combination of Eqs. (5) and (10) with the additional interaction term

$$S_{a_2} = \sum_{\mathbf{x}} (aa_2)\frac{1}{2}\,\mathrm{Tr}(a\Phi^\dagger\Phi)\,\mathrm{Tr}(a\Sigma^2). \tag{12}$$

3 Local update algorithms

In scalar theories, the common procedure for updating fields seems to be $4-5\times$overrelaxation, followed by a Metropolis update to obtain ergodicity. Gauge fields are updated using heat bath algorithms. We adapt this procedure, and the details are given in this section. An important thing to keep in mind is the fact that for flat potentials (especially Higgs in the broken phase), substantial improvement can be obtained by optimizing the updates for radial parts of the scalar fields (e.g. $\Phi = RV$, with $R \geq 0, V \in SU(2)$).

As a side note, we point out that the global radial update for the Higgs advertised in [3] is most likely not worth implementing, because it scales very badly with volume. Kari also ditched it already a long ago.

3.1 Overrelaxation for the scalars

Consider first a simple overrelaxation update based on Gaussian approximation of the scalar potential. The local Higgs potential can be written as

$$V(\Phi(x)) = s_a\phi_a + BR^2 + CR^4, \tag{13}$$

where the components ϕ_a are as defined in Eq. (3), $R^2 = \frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger\Phi$ and s_a is the hopping term staple. Neglecting the R^4 term for now, it is straightforward to overrelax ϕ_a by solving the new field components ϕ'_a from

$$\phi'_a + \frac{s_a}{2B} = -(\phi_a + \frac{s_a}{2B}). \tag{14}$$

To account for the quartic term, we simply do an accept/reject based on the difference $C(R')^4 - CR^4$. If C is small, the acceptance rate is high and the algorithm can be expected to work well. In my tests in the SM case $x = 0.034$ (very small C) I obtained acceptance of $\sim 95\%$. I believe Guy Moore used this update for SM nucleation. Note that this update generalizes very simply to e.g. adjoint scalars.

We can, however, do better by adapting the overrelaxation presented in [3] (has slight error in Y^2 parts) and [2] (no error). Because the algorithm is described in detail in the aforementioned references, I will not bother writing it here. The basic idea nevertheless is to perform separate updates on Cartesian parametrization of the Higgs field, which also is very efficient in updating the radial part. This update yields acceptance rate of whopping 99.8% or better, and I have tested that it facilitates tunneling much more effectively than the Gaussian approximation above, at least in the SM $x = 0.034$ case. **I use this optimized update for the Higgs**. (slightly confusing thing here: in Kari's code he does not seem to overrelax the Y component at all, or at least this is not very obvious. Perhaps updating the X part really is enough in practice?)

3.2 SU(2) gauge link heat bath

An efficient update for the SU(2) gauge links is the standard Kennedy-Pendleton heatbath [5]. The logic is as follows:

- For a link $U_i(x)$, write the local action in the form $S[U_i(x)] = \mathrm{Re}\,\mathrm{Tr}\,U_i(x)\hat{S}$, where $\hat{S}$ is a local staple (for SU(2), the trace here is automatically real). The staple is calculated from the Wilson action, as well as from hopping terms if scalars are present. In SM, it is given by

$$\hat{S} = -\frac{1}{2}\beta_G \sum_{j\neq i} \Big[U_j(x+i)U_i^\dagger(x+j)U_j^\dagger(x) \\ + U_j^\dagger(x+i-j)U_i^\dagger(x-j)U_j(x-j)\Big] - \Phi^\dagger(x)\Phi(x+i). \qquad (15)$$

- Normalize $\hat{S}$ to produce an SU(2) matrix: $V = -\hat{S}/k$ where $k = \sqrt{\det \hat{S}}$. Accounting for the minus sign, the distribution to be generated for U is $\sim \exp[k\,\mathrm{Re}\,\mathrm{Tr}\,UV]dU$. Because the Haar measure stays invariant under right multiplication with V^{-1}, the distribution for UV^{-1} is just

$$\sim \exp[k\,\mathrm{Re}\,\mathrm{Tr}\,U]dU = \exp[2ka_0]\frac{1}{2\pi^2}\delta(a^2-1)d^4a, \qquad (16)$$

where we wrote $U = a_0\sigma_0 + ia_i\sigma_i$.

- Now we just generate a_0 and a_i using the recipe described in [5] to get a new U', but since this new link follows the distribution of UV^{-1} we need to multiply by $V^{-1} = V^\dagger$. Thus the new link variable is $U'V^\dagger$.

If an adjoint scalar is present, this simple recipe no longer works because the adjoint hopping term is quadratic in $U_i(x)$. Instead, we generate a new link candidate following the above recipe, and then accept it with the probability $\min[1, \exp(H - H')]$, where H and H' are the adjoint hopping terms before and after the link update. Acceptance rate is typically good, $\sim 90\%$.

4 Sample histograms and comparison to old results

4.1 SU(2)+Higgs in 3d

The SM case can be compared to Ref. [3]. Note that their histograms are given in terms of R^2, which is related to our Φ as

$$R^2 = \frac{2}{\beta_H}\frac{1}{2}\text{Tr}(\Phi^\dagger\Phi)_{\text{latt}} = \frac{2a}{\beta_H}(\phi^\dagger\phi)_{\text{cont}} \tag{17}$$

and β_H can be calculated from given continuum parameters via their Eqs. (2.5)-(2.7). In their histograms they seem to give the pure lattice value for R^2, i.e. no divergence has been subtracted. **Our histograms are also presented in terms of the lattice fields.** Note that they parameterize their x, y in terms of T^* and m_H^*.

For the histogram in Fig. 6 of Ref. [3] (left figure), our lattice inputs corresponding to $\beta_G = 8, m_H^* = 35$ GeV, $T^* = 94.181$ GeV are

$$\lambda = 0.00915047 \quad m_\phi^2 = -0.207404. \tag{18}$$

Some histograms produced with my code using these inputs are shown in Fig. 3.

Note that David's code takes in $g = 0.707107$ instead of β_G. Note also that David's conversion.py script defined $ag^2 = 2/\beta_G$ instead of the usual $ag^2 = 4/\beta_G$, and correspondingly the Wilson term was normalized differently in the code. Now that we use the standard convention, the conversion script is outdated.

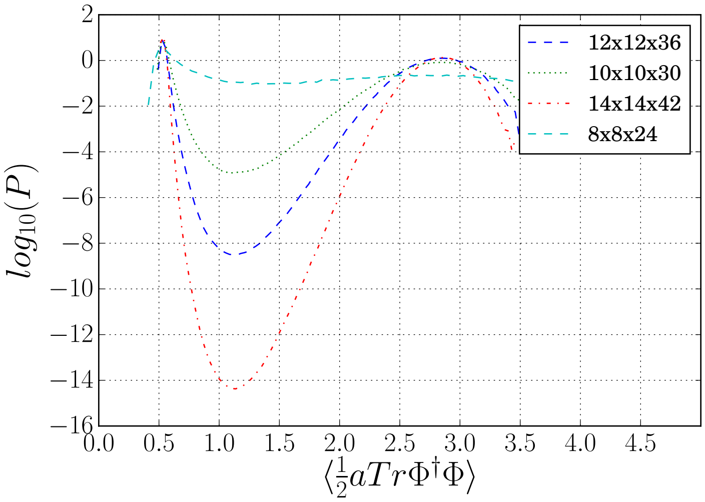

Figure 3: Histograms for the SM case with inputs as in Eq. (18). The smallest volume was done without multicanonical. The histograms agree well with those in Fig. 6 of Ref. [3], although our normalization is still off here. Their $\beta_H = 0.3450806$ here, so converting their x-axis to ours using Eq (17) we see that the peaks are located exactly as in their figure.

For the right-hand side histogram in Fig. 6 of Ref. [3], the inputs corresponding to $\beta_G = 8, m_H^* = 70$ GeV, $T^* = 153.620$ GeV are

$$\lambda = 0.0448519 \quad m_\phi^2 = -0.287506. \tag{19}$$

For David's code again $g = 0.707107$.

4.2 SU(2)+adjoint scalar in 3d

We can compare to histograms in Fig 4 of Ref. [4]. In this case Σ^a is identified as the temporal scalar A_0^a originating from thermal screening of the gauge field A_μ^a. They define

$$y = \frac{m_D^2(g^2)}{g^4}, \quad x = \frac{\lambda_A}{g^2} = \frac{b_4}{g^2}, \tag{20}$$

in terms of which they give their results (these are in $\overline{\text{MS}}$!). In Fig. 4 they give x, β_G and volume for the histograms, while y_c needs to be calculated from their Eq (5.12).

show histograms but only give the critical y_c for the case $x = 0.20$, in Fig. 2. Their volume is also quite large, 28^3, and we need multicanonical in order to simulate this volume. Nevertheless, reading off the critical y corresponding to this volume and $\beta_G = 16$ from their Fig. 2, we have $y_c \approx 0.154$. This fixes the continuum mass μ_Σ^2 in our parameterization.

Their volume in the $x = 0.20$ histogram is 28^3 and the mixed-phase suppression quite strong, so we need a working multicanonical update for the adjoint to compare this case. With just standard metropolis we may nevertheless use smaller volume, 8^3, and preliminary simulations are in some agreement with their critical values.

Applying this, we obtain lattice parameters for David's code corresponding to their $x = 0.20$ critical histogram:

$$g_{\text{latt}} = 0.5, \quad b_{4,\text{latt}} = 0.05, \quad m^2_{\Sigma,\text{ latt}} = -0.315315. \tag{21}$$

Histograms in Ref. [4] are given in terms of $A_0^a A_0^a/(2g^2)$ (I don't know if this is the lattice A_0^a or not!!). The condensate discontinuity in their histograms is related to David's histograms by

$$\Delta\langle a\Sigma_0^a\Sigma_0^a\rangle_{\text{David}} = \frac{8}{\beta_G}\Big(\frac{\Delta\langle A_0^a A_0^a\rangle}{2g^2}\Big)_{\text{Kari}}. \tag{22}$$

4.3 Comparing to Kari's simulations in Higgs + A_0 theory

We can compare the full fundamental + adjoint case to the histograms in Fig. 23 of Ref. [3], with Σ again being the A_0 field. Our DR suggests that the parameters need to be fixed (in terms of 4d parameters) as

$$\begin{aligned}\mu_\Sigma^2 =& m_D^2 = g_4^2 T^2\left(\frac{5}{6} + \frac{N_f}{3}\right), \quad b_4 = T\frac{g_4^4}{16\pi^2}\left(\frac{17+N_f}{3}\right), \\ a_2 =& 2h_3 = \frac{1}{2}g_4^2 T\Big[1 + \frac{1}{16\pi^2}\Big(\frac{17}{2} - 4(4\ln 2 - 1)\Big)g_4^2 - 6y_t^2 + 12\lambda_4\Big]\end{aligned} \tag{23}$$

These follow simply from 1-loop DR in the SM. However, this is not quite what they use in Ref. [3], instead they have made some approximations.

Converting their action to our notation, we see that their a_2-term reads

$$\sum_{\mathbf{x}} \frac{1}{4} a g^2 (aA_0^a A_0^a)(a\phi^\dagger\phi), \tag{24}$$

i.e. they fix $h_3 = \frac{1}{2}a_2 = \frac{1}{4}g_3^2$, which is just the leading-order result from DR.

Performing a similar conversion for the b_4 term, we find that their quartic coupling is

$$\beta_4^A = 4\frac{b_4}{g_3^4 a},$$

and judging from Ref. [6] it seems that they fix

$$b_4 \equiv \lambda_A = \frac{17 g_4^4 T}{48\pi^2}, \tag{25}$$

i.e. they ignore the fermionic contribution from DR. Similarly their mass in the continuum seems to be simply

$$m_D^2 = \frac{5}{6} g_4^2 T^2. \tag{26}$$

Note that all parameters in the A_0 sector are quite small, and the doublet will dominate the simulations. Nevertheless, to my knowledge there are no other simulations in the literature performed with both Higgs and an adjoint scalar.

Now for some complications: in Ref. [3] they do not actually list the explicit input parameters for the A_0 simulations. They they values for m_H^* and T^*, but only give the relations to x, y in terms of these **in the case of a fundamental Higgs only**, i.e. the relations already include the integration over A_0. So we need to figure out what they used for x, y in this case.

I asked Kari to search through his old files and he found the following inputs for the fundamental + adjoint case:

$$\beta_G = 12, \quad \beta_R = 0.00124, \quad \beta_2^A = 1.136, \quad \beta_4^A = 0.1914, \tag{27}$$

with only β_H varying. According to Ref. [3] the A_0 histograms there are evaluated at $\beta_H = 0.34771$. In terms of these, a straightforward conversion to our lattice parameters gives:

$$m_\phi^2 = 2\left(\frac{1}{\beta_H} - 3 - \frac{2\beta_R}{\beta_H}\right) = -0.262345 \tag{28}$$

$$\lambda = \frac{4\beta_R}{\beta_H^2} = 0.0410249 \tag{29}$$

$$m_\Sigma^2 = -\frac{2\beta_2^A}{\beta_G} = -0.189333 \tag{30}$$

$$b_4 = \frac{4\beta_4^A}{\beta_G^2} = 0.00531667 \tag{31}$$

$$a_2 = \frac{2}{\beta_G} = \frac{1}{6} = 0.16666667. \tag{32}$$

A Lattice-continuum relations

Due to renormalization scheme dependence, the relation between continuum and lattice parameter is nontrivial. In a superrenormalizable 3d theory, $\overline{\text{MS}}$ masses run at two loops while dimensionless couplings are RG invariant. Furthermore, the condensates such as $\langle \phi^\dagger \phi \rangle$ are scheme dependent and need to be matched, although their renormalization is additive and does not affect differences (which are sufficient for extracting the physics). In practice the condensates can be matched by calculating the unphysical zero-point divergence of the effective potential. The schemes can be matched exactly at two loop level, so that the lattice-continuum relations do not obtain further corrections at higher orders. This has been done in Refs. [7, 8] for the SM and some extensions. Here we collect the results relevant for our cases.

Denoting m_i = lattice mass, μ_i = $\overline{\text{MS}}$ mass, the two are related as

$$m_i^2 = \mu_i^2(\Lambda) + \delta m_i^2(\hbar) + \delta m_i^2(\hbar^2), \tag{33}$$

where $\hbar$ is the loop counting parameter. Constants often appearing in the counterterms read [8]:

$$\begin{aligned}\Sigma \approx& 3.17591153625, \quad \zeta \approx 0.08849, \quad \delta \approx 1.942130, \quad \rho \approx -0.313964, \\ \kappa_1 \approx& 0.958382, \quad \kappa_2 = \frac{\Sigma^2}{4} - \frac{\delta}{2} - \frac{1}{4}, \quad \kappa_3 \approx 0.751498, \quad \kappa_4 \approx 1.204295. \end{aligned} \tag{34}$$

We will denote the $\overline{\text{MS}}$ scale in 3d by Λ and the 3d parameters are assumed to be in natural 3d units (e.g. g^2 has dimension GeV).

A.1 SU(2) + fundamental Higgs

The relation between lattice and continuum Higgs condensates is

$$\langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \rangle_{\text{cont}} = \langle \phi^\dagger \phi \rangle_{\text{cont}} = \langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \rangle_{\text{latt}} + \frac{d(\delta V)}{dm_\phi^2}, \tag{35}$$

where

$$\delta V = -2m_\phi^2 \frac{\Sigma}{4\pi a} - \frac{3}{16\pi^2} g^2 m_\phi^2 \Big(\ln \frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta \Big). \tag{36}$$

The full mass counterterm is

$$\begin{aligned}(\delta m_\phi^2)_{\text{fund}} =& -(\frac{3}{2}g^2 + 6\lambda)\frac{\Sigma}{4\pi a} \\ &- \frac{1}{16\pi^2}\Big[\Big(\frac{51}{16}g^4 + 9\lambda g^2 - 12\lambda^2 \Big) \Big(\ln \frac{6}{a\Lambda} + \zeta \Big) + 9\lambda g^2 \Big(\frac{1}{4}\Sigma^2 - \delta \Big) \\ &+ \frac{3}{4} g^4 \Big(\frac{15}{16}\Sigma^2 + \frac{\pi}{3}\Sigma + \frac{5}{4} - \frac{7}{2}\delta - 4\rho + 4\kappa_1 - \kappa_2 - \kappa_3 - 3\kappa_4 \Big) \Big]. \end{aligned} \tag{37}$$

A.2 SU(2) + adjoint scalar

The triplet condensate, we have:

$$\frac{1}{2}\langle \Sigma^a \Sigma^a \rangle_{\text{cont}} = \langle \operatorname{Tr} \Sigma^2 \rangle_{\text{cont}} = a^{-1} \langle \operatorname{Tr} \Sigma^2 \rangle_{\text{latt}} + \frac{d(\delta V)}{dm_\Sigma^2}, \tag{38}$$

where the counterterm is the sum of

$$\begin{aligned}\delta V(\hbar) =& -\frac{3}{2} m_\Sigma^2 \frac{\Sigma}{4\pi a} \\ \delta V(\hbar^2) =& -\frac{1}{16\pi^2} 6 g^2 m_\Sigma^2 \Big(\ln \frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta \Big). \end{aligned} \tag{39}$$

The mass counterterm is the sum of

$$(\delta m_\Sigma^2(\hbar))_{\text{adj}} = -(4g^2 + 5b_4)\frac{\Sigma}{4\pi a}, \tag{40}$$

$$\begin{aligned}(\delta m_\Sigma^2(\hbar^2))_{\text{adj}} =& -\frac{1}{16\pi^2}\Big[\Big(20 b_4 g^2 - 10 b_4^2 \Big) \Big(\ln \frac{6}{a\Lambda} + \zeta \Big) + 20 b_4 g^2 \Big(\frac{1}{4}\Sigma^2 - \delta \Big) \\ &+ 2g^4 \Big(\frac{5}{4}\Sigma^2 + \frac{\pi}{3}\Sigma - 6\delta - 6\rho + 4\kappa_1 - \kappa_2 - \kappa_3 - 3\kappa_4 \Big) \Big]. \end{aligned} \tag{41}$$

Note that Ref. [7] writes the adjoint as $A_0 = A_0^a \sigma_a$, so their covariant derivative term has different factors of 2 than we do. Overall the actions are still equivalent. Their parameters are simply $m_D = m_\Sigma, \lambda_A = b_4$.

A.3 SU(2) + fundamental Higgs + adjoint scalar

The vacuum divergence is not modified by the a_2 term, so the counterterm relating the condensates is simply the sum of Eqs. (36) and (39):

$$\delta V = -(2m_\phi^2 + \frac{3}{2}m_\Sigma^2)\frac{\Sigma}{4\pi a} - \frac{3}{16\pi^2}g^2(m_\phi^2 + 2m_\Sigma^2)\Big(\ln\frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta\Big). \tag{42}$$

For the masses, we have

$$m_\phi^2 = \mu_\phi^2(\Lambda) + (\delta m_\phi^2)_{\text{fund}} + (\delta m_\phi^2)_{\text{both}} \tag{43}$$
$$m_\Sigma^2 = \mu_\Sigma^2(\Lambda) + (\delta m_\Sigma^2)_{\text{adj}} + (\delta m_\Sigma^2)_{\text{both}} \tag{44}$$

where the new contributions read

$$\begin{aligned}(\delta m_\phi^2)_{\text{both}} =& -\frac{3}{2}a_2\frac{\Sigma}{4\pi a} - \frac{1}{16\pi^2}\Big[\Big(-\frac{3}{4}g^4 + 6a_2g^2 - \frac{3}{2}a_2^2\Big)\Big(\ln\frac{6}{a\Lambda} + \zeta\Big) \\ &+ 6a_2g^2\Big(\frac{1}{4}\Sigma^2 - \delta\Big) - 3g^4\rho\Big],\end{aligned} \tag{45}$$

$$\begin{aligned}(\delta m_\Sigma^2)_{\text{both}} =& -2a_2\frac{\Sigma}{4\pi a} - \frac{1}{16\pi^2}\Big[\Big(-g^4 + 3a_2g^2 - 2a_2^2\Big)\Big(\ln\frac{6}{a\Lambda} + \zeta\Big) \\ &+ 3a_2g^2\Big(\frac{1}{4}\Sigma^2 - \delta\Big) - 4g^4\rho\Big].\end{aligned} \tag{46}$$

Note that there are also contributions not proportional to a_2.

B Code technicalities

- Dimensions of spacetime and the lattice size are not fixed beforehand; they are to be read in from config. Our lattice is a $L_1 \times L_2 \times \cdots \times L_n$ sized periodic hypercube. It is recommended to use even numbers for L_i; otherwise checkerboard updating is not well defined.

- We use a single index i to label lattice sites and use this index to access field values etc, instead of Cartesian $(x, y, z, \dots)$ coordinates (these are used in initial layouting). In short, sites are ordered so that the site at origin $(0, 0, 0, \dots)$ has $i = 0$ (”upper left corner”), next site in positive direction 1 has $i = 1$ etc. When we reach last site in direction 1, we move one step in direction 2 and repeat, and so on. Neighbor sites are stored in lookup tables before starting the simulation. **Update 23.8.2019:** Added routines in layout.c to further reorder the sites by parity. This allows for more optimized update sweeps (in quick test runs the improvement was 20%!). The full ordering using this option is: 1. even sites 2. odd sites 3. even halos 4. odd halos. To do: check if memory configuration could be optimized as well with parity ordering.

- Fields are dynamic arrays of doubles (double*, double**, double***) that are accessed as field[site index][direction][component] for gauge links, field[site index][component] for non-gauge fields. Gauge singlets are simply field[site]. Memory for fields is allocated contiguously.

- OpenMPI is used for parallelization. The lattice is split into hypercubes of equal sizes with side lengths L_i^{slice}, and these are then laid out based on their MPI ranks with same indexing logic as for lattice sites. We treat each node as being a hypercube with side lengths $L_i^{\text{slice}} + 2$, where the two extra sites are halos that need to be updated using MPI communications. Halo site indexes come after real sites, but otherwise follow same ordering logic. Due to periodicity, it can happen that the physical site corresponding to a halo is actually a real site in the same node. These "self halos" are removed by simply changing the neighbor lookup table to point to the real site instead. All this is implemented in layout.c.

- Communication between nodes is implemented in comms.c. We use a "comlist" structure to store information of what site indices our node is supposed to send to which nodes, and what halo site indices do we update with data received from them. Send/receive data is copied into temporary buffers each time we update halos. Each update, we first send our data to all neighbors using nonblocking sends, then proceed with blocking receives to update our halos, and finally wait for all of our sends to go through (usually done by the time we get to the wait loop). Variable c.comms_time keeps track of the time spent on MPI communications (excluding global multicanonical checks).

- Updating the lattice is done with checkerboard style sweeps (imagine a chessboard). We specify parity of a site to be EVEN if $x + y + z + \dots$ is an even number, ODD otherwise. When sweeping, we first update all sites with EVEN parity, including halos, and then repeat for ODD sites. Gauge links are updated one direction at a time, because the directions are not independent (think of the Wilson plaquette).

References

[1] B. A. Berg and T. Neuhaus, Phys. Lett. B **267** (1991) 249. doi:10.1016/0370-2693(91)91256-U

[2] M. Laine and K. Rummukainen, Nucl. Phys. B **535** (1998) 423 doi:10.1016/S0550-3213(98)00530-6 [hep-lat/9804019].

[3] K. Kajantie, M. Laine, K. Rummukainen and M. E. Shaposhnikov, Nucl. Phys. B **466** (1996) 189 doi:10.1016/0550-3213(96)00052-1 [hep-lat/9510020].

[4] K. Kajantie, M. Laine, K. Rummukainen and M. E. Shaposhnikov, Nucl. Phys. B **503** (1997) 357 doi:10.1016/S0550-3213(97)00425-2 [hep-ph/9704416].

[5] A. D. Kennedy and B. J. Pendleton, Phys. Lett. **156B** (1985) 393. doi:10.1016/0370-2693(85)91632-6

[6] K. Farakos, K. Kajantie, K. Rummukainen and M. E. Shaposhnikov, Nucl. Phys. B **425** (1994) 67 doi:10.1016/0550-3213(94)90173-2 [hep-ph/9404201].

[7] M. Laine, Nucl. Phys. B **451** (1995) 484 doi:10.1016/0550-3213(95)00356-W [hep-lat/9504001].

[8] M. Laine and A. Rajantie, Nucl. Phys. B **513** (1998) 471 doi:10.1016/S0550-3213(97)00709-8 [hep-lat/9705003].