Miscellaneous documentation

Lauri Niemi

March 1, 2024

0 Preface

These are miscellaneous notes explaining implementation details etc technical things. The contents are not necessarily up to date with what the code does.

1 Multicanonical algorithm

In a first-order phase transition, mixed-phase configurations are exponentially suppressed by a factor proportional to system size. 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. Remember, however, that the simulation is formally correct with any W, as reweighting "removes" the weight in the end. **In the measurement file, we measure $-W(R)$ instead of $W(R)$.**

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.

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 [3]:

- Perform an even-odd sweep on the lattice, updating half of the sites locally using *canonical* updates such as standard 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 rates reported in [3, 2].

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 old lecture notes by Kari Rummukainen.

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). This method generally works well if the mixed-phase suppression is strong.

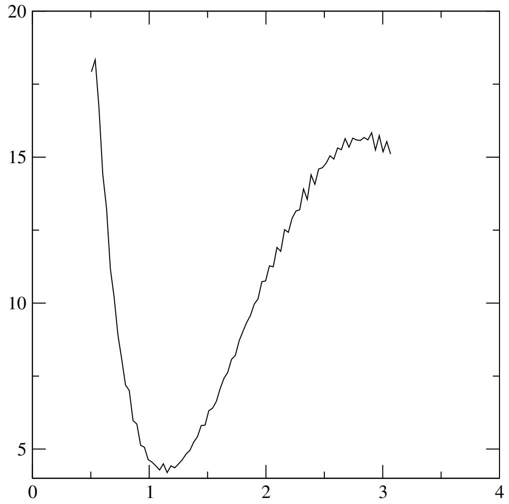


Figure 1: Sample weight function produced using the above algorithm. This corresponds to the SM case with $m_H^* = 35$ GeV, $\beta_G = 8$ and $V = 10 \times 10 \times 30$. Horizontal axis is the multicanonical order parameter $\langle \frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger\Phi \rangle$.

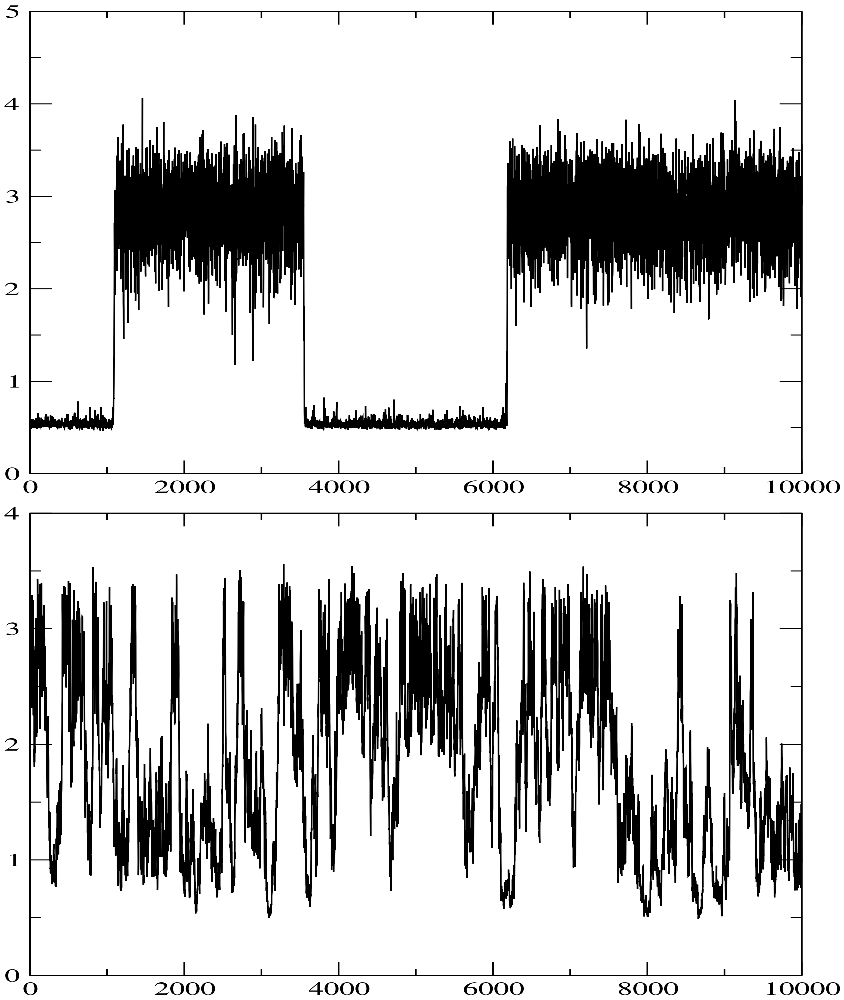


Figure 2: Time histories for the observable $\langle \frac{1}{2}\,\mathrm{Tr}\,\Phi^\dagger\Phi \rangle$, top: no multicanonical weight, bottom: using the weight in Fig. 1. Without the weight function, the system is stuck in the pure phases and only rarely samples the mixed-phase configurations. Here 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, which 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. All examples here are for 3D theories.

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. If the vector $\phi = \frac{1}{\sqrt{2}}(H_1, H_2)^T$, the matrix field is obtained as

$$\Phi = \frac{1}{\sqrt{2}}\begin{pmatrix} H_2^* & H_1 \\ -H_1^* & H_2 \end{pmatrix} \equiv \frac{1}{\sqrt{2}}(h_0\sigma_0 + ih_k\sigma_k), \tag{3}$$

and the code uses the RHS components h_0, h_k. They are related to the vector components as

$$h_0 = \operatorname{Re} H_2, \quad h_1 = \operatorname{Im} H_1, \quad h_2 = \operatorname{Re} H_1. \quad h_3 = -\operatorname{Im} H_2. \tag{4}$$

This can be checked by comparing hopping terms in the two parametrizations as the ordering does make a difference. Note that $\frac{1}{2}\operatorname{Tr}\Phi^\dagger\Phi = \phi^\dagger\phi = \frac{1}{2}\phi_\mu\phi_\mu$.

The discretized action reads (i, j denote directions)

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

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 $P_{ij}(x) = \operatorname{Re}\operatorname{Tr} U_i(x)U_j(x+i)U^\dagger_i(x+j)U^\dagger_j(x)$. 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 $\frac{1}{2}a \operatorname{Tr} \Phi^\dagger \Phi$ etc, and takes as inputs the combinations $m_{\text{lat}}^2 \equiv a^2 m_\phi^2$, $\lambda_{\text{lat}} \equiv a\lambda$ 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.

2.1.1 Notes on multiple Higgs doublets

Suppose we have two doublets ϕ_1, ϕ_2. There are four independent bilinears: $\phi_{ij} = \phi_i^\dagger \phi_j$, and the potential used in code is

$$\begin{aligned} V(\phi_1, \phi_2) =& m_1^2 \phi_{11} + m_2^2 \phi_{22} + \frac{1}{2}(m_{12}^2 \phi_{12} + h.c.) + \lambda_1 \phi_{11}^2 + \lambda_2 \phi_{22}^2 + \lambda_3 \phi_{11}\phi_{22} + \lambda_4 \phi_{12}\phi_{21} \\ &+ \frac{1}{2}(\lambda_5 \phi_{12}^2 + \lambda_6 \phi_{11}\phi_{12} + \lambda_7 \phi_{22}\phi_{21} + h.c.). \end{aligned} \tag{9}$$

(note: factor of 1/2 in all complex terms!!)

In terms of matrices Φ_1, Φ_2 defined as in eq. (3), we have

$$\phi_{ii} = \frac{1}{2} \operatorname{Tr} \Phi_i^\dagger \Phi_i, \quad R \equiv \operatorname{Re} \phi_{12} = \frac{1}{2} \operatorname{Tr} \Phi_1^\dagger \Phi_2, \quad I \equiv \operatorname{Im} \phi_{12} = -i \frac{1}{2} \operatorname{Tr} \Phi_1 \sigma_3 \Phi_2^\dagger \tag{10}$$

and the potential can be written as

$$\begin{aligned} V =& m_1^2 \phi_{11} + m_2^2 \phi_{22} + \operatorname{Re} m_{12}^2 R - \operatorname{Im} m_{12}^2 I + \lambda_1 \phi_{11}^2 + \lambda_2 \phi_{22}^2 + \lambda_3 \phi_{11}\phi_{22} + \lambda_4 (R^2 + I^2) \\ &+ \operatorname{Re} \lambda_5 (R^2 - I^2) - 2 \operatorname{Im} \lambda_5 RI + \phi_{11}(\operatorname{Re} \lambda_6 R - \operatorname{Im} \lambda_6 I) + \phi_{22}(\operatorname{Re} \lambda_7 R + \operatorname{Im} \lambda_7 I), \end{aligned} \tag{11}$$

which is easy to program in.

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

$$\mathcal{L} = \frac{1}{4}(F_{ij}^a)^2 + \frac{1}{2}(D_i \Sigma^a)^2 + \frac{1}{2}\mu_\Sigma^2 \Sigma^a \Sigma^a + \frac{1}{4} b_4 (\Sigma^a \Sigma^a)^2. \tag{12}$$

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

$$\begin{aligned} S_{\text{latt}} =& \beta_G \sum_{\mathbf{x}, i<j} \Big(1 - \frac{1}{2} P_{ij}(x)\Big) + 2 \sum_{\mathbf{x}, i} \Big(a \operatorname{Tr} \Sigma(x)^2 - a \operatorname{Tr} \Sigma(x) U_i(x) \Sigma(x+i) U_i^\dagger(x)\Big) \\ &+ \sum_{\mathbf{x}} \Big[(am_\Sigma)^2 (a \operatorname{Tr} \Sigma^2) + ab_4 (a \operatorname{Tr} \Sigma^2)^2\Big], \end{aligned} \tag{13}$$

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

$$\begin{aligned}\mathcal{L} =& \frac{1}{4}(F_{ij}^a)^2 + (D_i\phi)^\dagger(D_i\phi) + \mu_\phi^2\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2 \\ &+ \frac{1}{2}(D_i\Sigma^a)^2 + \frac{1}{2}\mu_\Sigma^2\Sigma^a\Sigma^a + \frac{1}{4}b_4(\Sigma^a\Sigma^a)^2 + \frac{1}{2}a_2\phi^\dagger\phi\Sigma^a\Sigma^a,\end{aligned} \tag{14}$$

and the lattice action is simply a combination of Eqs. (5) and (13) with the additional interaction term

$$S_{a_2} = \sum_{\mathbf{x}}(aa_2)\frac{1}{2}\,\mathrm{Tr}(a\Phi^\dagger\Phi)\,\mathrm{Tr}(a\Sigma^2). \tag{15}$$

2.4 Adding the $\mathrm{U(1)_Y}$ subgroup

Simulations in the electroweak SU(2) × U(1) effective theory in 3d were discussed in [10]. Including the U(1) gauge field to the actions above is straightforward, but note that non-perturbatively there is a difference between "compact" (gauge group really is U(1)) and "non-compact" (gauge group is real numbers under addition) formulations.

The link variable is just a complex number $u_i(x) = \exp(i\alpha_i(x))$; the simulation will operate on the real numbers $\alpha_i(x)$ (in code: u1field[x][dir]). The plaquette trace for a "fundamental" U(1) Wilson line is as

$$p_{ij}(x) = \exp\{i[\alpha_i(x) + \alpha_j(x+i) - \alpha_i(x+j) - \alpha_j(x)]\} \equiv \exp[if_{ij}(x)]. \tag{16}$$

As in [10] (and in the book by Jan Smit) we can consider actions where the plaquette is in an arbitrary irreducible representations of U(1). The reps can be labeled by an integer number r. A possible action is then

$$S_{\mathrm{U}(1)} = \beta_G' \sum_{x,i<j}\left[1 - \frac{1}{2}\Big(p_{ij}^r + (p_{ij}^*)^r\Big)\right] = \beta_G' \sum_{x,i<j}\Big[1 - \cos(rf_{ij})\Big]. \tag{17}$$

Taking $\beta_G' \propto 1/r^2$, any r gives the same naive continuum limit. In compact formulation, r needs to be an integer and the gauge transformations operate on the $a_i(x)$ modulo (2π), *i.e.* in the simulations too we can restrict $-\pi < a_i(x) \le \pi$. Formally $r \to 0$ corresponds to the non-compact formulation.

In the compact theory, matter fields need to be in representations of U(1) and thus only integer charges are allowed. Of course, in continuum theory the charges can be normalized at will by rescaling the gauge coupling, in particular the Higgs covariant derivative in usual normalization is

$$D_i\phi = \partial_i\phi + \frac{1}{2}igA_i^a\sigma_a + Yig'B_i \tag{18}$$

with $Y = 1/2$. The Higgs doublet Φ, in matrix parametrization, transforms under a SU(2) × U(1) gauge transformation as

$$\Phi(x) \to G(x)\Phi(x)e^{-iY\alpha(x)\sigma_3}, \tag{19}$$

where σ_3 is the third Pauli matrix (to check that this is the correct transformation rule, apply $\phi \to e^{iY\alpha}G(x)\phi$ on the vector ϕ and see how the elements of Φ transform).

It is best to rescale the charge, $\tilde{g}' \equiv Yg'$, and write the lattice action in normalization where Higgs is in $r = 1$ irrep. Writing $u_i(x) = e^{ia\tilde{g}'B_i(x)}$, we find the correct continuum limit for the gauge action if

$$\beta_G' = \frac{1}{a\tilde{g}'^2r^2}. \tag{20}$$

Note that our renormalization scheme and DR gives g' instead of $\tilde{g}'$, but the conversion is trivial.

With the lattice gauge coupling defined as above, the lattice covariant derivative term is

$$\sum_{x,i} \Big[\operatorname{Tr} \Phi^\dagger(x)\Phi(x) - \operatorname{Re} \operatorname{Tr} \Phi^\dagger(x) U_i(x) \Phi(x+i) e^{-i\alpha_i(x)\sigma_3} \Big]. \tag{21}$$

To calculate this, it is often convenient to write

$$e^{-i\alpha\sigma_3} = \sigma_0 \cos(\alpha) - i\sigma_3 \sin(\alpha), \tag{22}$$

with σ_0 again being the 2×2 identity matrix.

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, and it does not seem to be included in more recent works of the authors.

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

$$V(\Phi(x)) = s_a\phi_a + BR^2 + CR^4, \tag{23}$$

where the components ϕ_a are as defined in Eq. (3) (TODO fix notation), $R^2 = \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi$ 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

$$\phi'_a + \frac{s_a}{2B} = -(\phi_a + \frac{s_a}{2B}). \tag{24}$$

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 $\sim 95\%$. 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 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 both Higgs and the adjoint scalar**.

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

$$\begin{aligned}\hat{S} = &-\frac{1}{2}\beta_G \sum_{j\neq i} \Big[U_j(x+i)U_i^\dagger(x+j)U_j^\dagger(x) \\ &+ U_j^\dagger(x+i-j)U_i^\dagger(x-j)U_j(x-j)\Big] - \Phi(x+i)e^{-i\alpha_j(x)\sigma_3}\Phi^\dagger(x). \end{aligned} \tag{25}$$

  Note that the U(1) field does contribute to the hopping term. In code it's useful to write $e^{-i\alpha_j(x)\sigma_3} = \cos\alpha_j(x)\sigma_0 - i\sin\alpha_j(x)\sigma_3$.

- 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

$$\sim \exp[k\,\mathrm{Re}\,\mathrm{Tr}\,U]dU = \exp[2ka_0]\frac{1}{2\pi^2}\delta(a^2-1)d^4a, \tag{26}$$

  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, $> 80\%$.

4 Sample histograms and comparisons 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

$$R^2 = \frac{2}{\beta_H}\frac{1}{2}\mathrm{Tr}(\Phi^\dagger\Phi)_{\mathrm{latt}} = \frac{2a}{\beta_H}(\phi^\dagger\phi)_{\mathrm{cont}} \tag{27}$$

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

$$\lambda = 0.00915047 \quad m_\phi^2 = -0.207404. \tag{28}$$

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

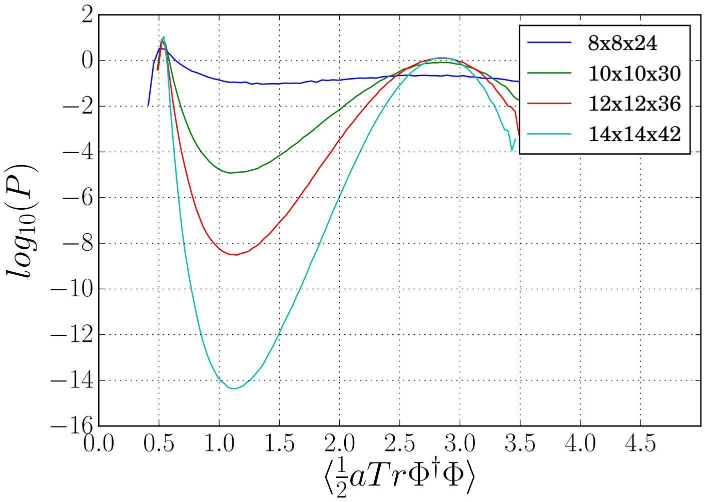


Figure 3: Histograms for the SM case with inputs as in Eq. (28). The smallest volume was done without multicanonical. The histograms agree well with those in Fig. 6 of Ref. [3], although our normalization is still off here. Their $\beta_H = 0.3450806$ here, so converting their x-axis to ours using Eq (27) we see that the peaks are located exactly as in their figure.

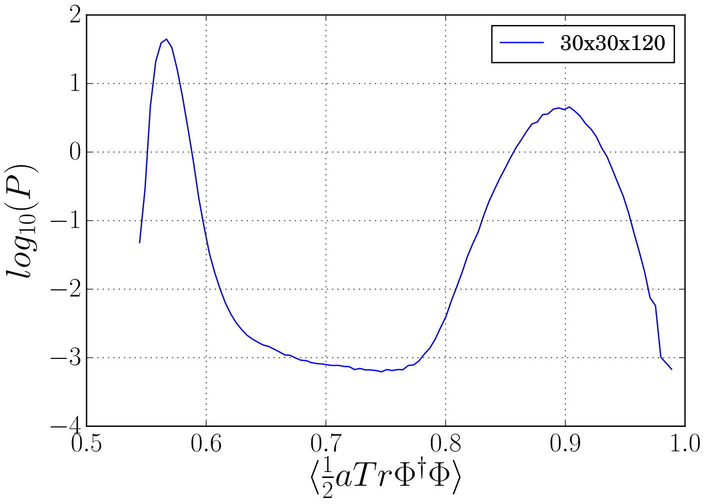


Figure 4: Histogram corresponding to $m_H^* = 60$ GeV, $T^* = 137.669$ GeV, $\beta_G = 8$. Compare with Figure 5 in Ref. [3]. Our lattice inputs here are $m_\phi^2 = -0.263633, \lambda = 0.0322228$.

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

$$\lambda = 0.0448519 \quad m_\phi^2 = -0.287506. \tag{29}$$

Haven't really tested these yet because the other cases worked so well...

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

$$y = \frac{m_D^2(g^2)}{g^4}, \quad x = \frac{\lambda_A}{g^2} = \frac{b_4}{g^2}, \tag{30}$$

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). The lattice mass is then obtained from y_c by subtracting the lattice divergence (Appendix A).

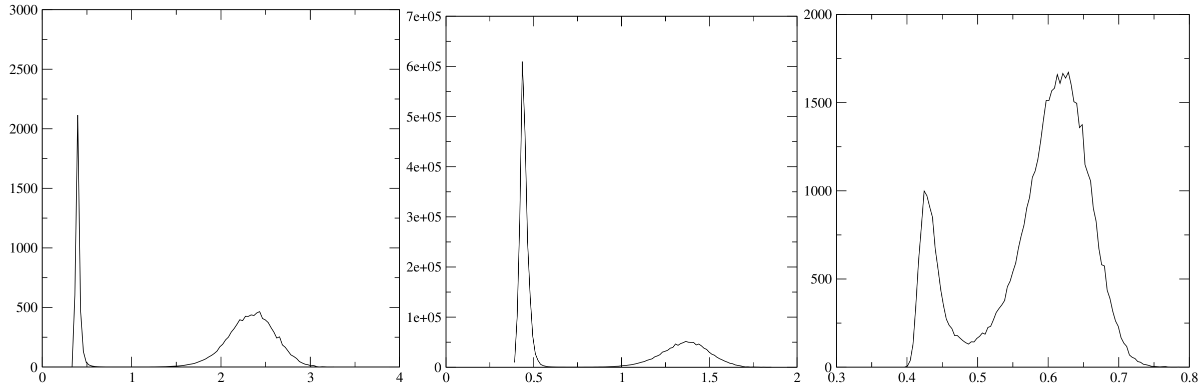
Some lattice inputs corresponding to histograms in their Fig 4 are:

$$x = 0.04, \beta_G = 32 : m_\Sigma^2 = -0.123686, b_4 = 0.005 \tag{31}$$

$$x = 0.12, \beta_G = 8 : m_\Sigma^2 = -0.54502, b_4 = 0.06 \tag{32}$$

$$x = 0.20, \beta_G = 16 : m_\Sigma^2 = -0.315858, b_4 = 0.05. \tag{33}$$

Histograms corresponding to these are shown in Fig 5.



(a) $x = 0.04, V = 18^3$

(b) $x = 0.12, V = 12^3$

(c) $x = 0.20, V = 28^3$

Figure 5: Histograms obtained with the inputs of Eq. (31). Horizontal axis is $\langle \mathrm{Tr}\,\Sigma^2 \rangle$ in lattice regularization. (a) and (b) were obtained with multicanonical weighting. In Ref. [4] they give the histograms in Fig 4 in terms of the continuum condensate $\langle \frac{A_0^a A_0^a}{2g_3^2} \rangle$. Because the condensates in $\overline{\text{MS}}$ and on the lattice are related by additative renormalization, the relation between *differences* in our and their histograms is $\Delta\langle \frac{A_0^a A_0^a}{2g_3^2} \rangle_{\text{Kari}} = \frac{\beta_G}{4} \Delta \langle \mathrm{Tr}\,\Sigma^2 \rangle_{\text{latt}}$. It is straightforward to check that the separation of the pure phases in our histograms matches what they have. Note that (c) does NOT give the same relative probabilities as in Ref. [4]. A possible explanation is that since their fit for $y_c(x)$ is performed at $V \to \infty$ limit, my value for the mass parameter may be slightly off here.

5 Measuring physical stuff

Warning. The following can contain errors in the details. If you're reading this in 2024 or later, see my recent papers instead.

5.1 Reweighting

Suppose the action depends on some variable x, and consider a canonical probability distribution of some quantity O evaluated at fixed x:

$$p_x(O) \propto e^{-S(x)}. \tag{34}$$

Suppose we then want to study the system with a slightly different value x'. Expanding S around this point we have

$$S(x) = S(x') + \frac{dS}{dx}\bigg|_{x=x'}(x-x') + \mathcal{O}(x-x')^2, \tag{35}$$

and in practice we can often approximate the derivative to be constant, at least if the values of x are not very far from each other: $\frac{dS}{dx}\Big|_{x=x'}(x-x') \equiv k = \text{const}$.

Then

$$p_x(O) \propto e^{-S(x')}e^{-k(x-x')} = Cp_{x'}(O)e^{-k(x-x')}. \tag{36}$$

The expectation value is

$$\langle O\rangle_x = \frac{1}{Z_x}\int D\phi\; Op_x(O) = \frac{Z_{x'}}{Z_x}C\langle Oe^{-k(x-x')}\rangle_{x'}, \tag{37}$$

and taking $O = 1$ fixes the proportionality constant to be

$$\frac{Z_{x'}}{Z_x}C = \frac{1}{\left\langle e^{-k(x-x')}\right\rangle_{x'}}. \tag{38}$$

So we have the reweighting formula: If a canonical expectation value is known at value x', at some other value x the expectation value can be obtained from

$$\langle O\rangle_x = \frac{\langle Oe^{-k(x-x')}\rangle_{x'}}{\langle e^{-k(x-x')}\rangle_{x'}} \tag{39}$$

provided that x is not too far from the original x'. In Monte Carlo,

$$\langle O\rangle_x = \frac{1}{Z_x}\sum_{\{\phi\}} O(\phi)e^{-S} \approx \frac{1}{N_\text{meas}}\sum_{i=1}^{N_\text{meas}} O_i, \tag{40}$$

where we sum over measurements on the right-hand side. Hence the reweighting formula becomes

$$\langle O\rangle_x \approx \frac{\sum_i O_i e^{-(x-x')k_i}}{\sum_i e^{-(x-x')k_i}}. \tag{41}$$

Reweighting with respect to the temperature. Now

$$\langle O\rangle_T \approx \frac{\sum_i O_i e^{-(T-T')k_i}}{\sum_i e^{-(T-T')k_i}}. \tag{42}$$

and on the lattice

$$k = \frac{dS}{dT} = \sum_x \frac{d}{dT}\left(m^2\frac{1}{2}\operatorname{Tr}\Phi^\dagger\Phi + \dots\right) = \sum_x\left(\frac{dm^2}{dT}\frac{1}{2}\operatorname{Tr}\Phi^\dagger\Phi + \dots\right). \tag{43}$$

Why does the derivative not act on the fields, even though our lattice field is clearly scaled by T to make it dimensionless? The reason is that the fields are just integration variables in the path integral, which should anyway be evaluated at a fixed temperature. The scaling of fields affects the couplings, and this effect is captured in the above expression (I haven't come up with a better argument...).

Now in a simulation we measure volume averages such as

$$\left\langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \right\rangle_i = \frac{1}{N} \sum_x \frac{1}{2} (\operatorname{Tr} \Phi^\dagger \Phi)_i. \tag{44}$$

Here the index i refers to a fixed configuration (or measurement) and N is the number of lattice sites, so that the physical volume is $V = a^3 N$. The "reweighting factor" k_i can therefore be written in terms of observables as

$$k_i = N \left(\frac{dm^2}{dT} \left\langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \right\rangle_i + \dots \right). \tag{45}$$

So to summarize, temperature reweighting can be once the T dependence of all couplings appearing in the (lattice) action is known and volume averages of all operators in the action are measured. In practice, the derivatives can be evaluated at the original temperature T' and approximated as constants.

Note that in statistical mechanics, the canonical probability weight is $e^{-\beta E} = e^{-E/T}$, so reweighting with respect to T would be performed using the energy. In our case the energy would be (from standard statistical physics)

$$E = T^2 \frac{\partial \ln Z}{\partial T} = -\frac{T^2}{Z} \int D\phi \, e^{-S} \frac{\partial S}{\partial T} = -T^2 \left\langle \frac{\partial S}{\partial T} \right\rangle. \tag{46}$$

In terms of Monte Carlo measurements:

$$E = -T^2 \frac{1}{N_{\text{meas}}} \sum_i^{N_{\text{meas}}} \frac{\partial S_i}{\partial T}, \tag{47}$$

which is not terribly useful because in reweighting, we need the energy at each measurement (but can be used for determining latent heat).

5.2 Latent heat

Latent heat is defined as the change in energy density across a phase transition. I can come up with two ways of measuring it from simulations (that hopefully should agree).

Option 1. As discussed above, the total energy for our system is given by

$$E = -T^2 \left\langle \frac{\partial S}{\partial T} \right\rangle, \tag{48}$$

which is now an average over Monte Carlo measurements. On the other hand,

$$\frac{\partial S_i}{\partial T} = N \left(\frac{dm^2}{dT} \left\langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \right\rangle_i + \dots \right) \tag{49}$$

is constructed from volume averages of local operators (which is what we measure). Note that this is the same quantity that we use for reweighting.

Now, it of course makes no sense to average over all configurations when looking for energy discontinuities. Instead we want to find the energies of the bulk phases and subtract them, so we should look at the histogram of $\partial S_i / \partial T$. Recipe for finding the latent heat is then to calculate the condensates of relevant operators in the bulk phases (histogram peaks), and take the difference. So the latent heat over T^4 (to make it dimensionless) is

$$\begin{aligned} \frac{L}{T^4} =& \frac{1}{T^4} \Delta \frac{E}{V} = -\frac{T^2}{T^4 a^3 N} \Delta \left\langle \frac{\partial S}{\partial T} \right\rangle \\ =& -\frac{1}{T^2 a^3 N} \Delta \underbrace{N \left(\frac{dm^2}{dT} \left\langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \right\rangle + \dots \right)}_{\text{"reweight string"}}, \end{aligned} \tag{50}$$

where $\Delta\langle\ldots\rangle = \langle\ldots\rangle_{\text{low-}T\text{ phase}} - \langle\ldots\rangle_{\text{high-}T\text{ phase}}$.

5.3 Finding the critical temperature without sampling pure phases

For strong transitions with strongly metastable phases (especially 2-steps) it can be difficult to find a good starting temperature for multicanonical sampling. This is because it is then practically impossible to get the system to tunnel even once near T_c, and tunneling may occur only at temperatures much lower than the critical temperature. In this case it is convenient to use the following method, which only samples mixed-phase configurations.

The recipe is as follows:

1. Use a long lattice, $n_z \gg n_x, n_y$, and prepare the system in a mixed configuration where sites with $z < n_z/2$ are in phase 1 and sites with $z > n_z/2$ are in phase 2.

2. For a suitable order parameter (*e.g.* $\langle \phi^\dagger\phi - \mathrm{Tr}\,\Sigma^2 \rangle$), use a multicanonical weight to constrain the system to a very narrow range around the starting value. In practice, choose the order parameter so that it starts with value 0, then set zero weight in, for example, range $(-0.1, 0.1)$ and infinite (=absurdly large) weight elsewhere (keep the weight constant during the simulation!). This corresponds to sampling the mixed-phase configurations only, because the weight prevents the system from evolving into a pure phase. On a long lattice, the phase boundaries (2 on a periodic lattice) stay perpendicular to the z-axis, and at critical temperature both phases should occupy half of the lattice.

3. Look at the histogram of the order parameter in the range of interest, and find the temperature where the histogram is flat. Easily done by reweighting.

Once the T_c is known, latent heat can be found simply by preparing the system in a pure phase and finding the expectation values of condensates there, and then repeating for the other phase. Getting the surface tension without actually studying tunneling is less straightforward, but Kari had some trick in mind for this too.

Some practical notes:

- If the system manages to escape outside the weighting range, there is nothing stopping it from getting stuck in a pure phase. Lesson: either use steep linear weight a la Kari that forces the system to "roll" back near the initial value if it attempts to escape, or just make sure that the "infinite" weight really is practically infinite.

- For thermalization, it (probably) helps to allow for small initial fluctuations when setting up the wall. However, in this case it's better to use a "linearly infinite" weight function (like Kari did) because the initial order parameter is now slightly random.

- If the temperature is far from T_c, it is not rare to find the system occupying only one end of the weight range.

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 (in the continuum limit!) 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. I have carefully checked that the expressions below match those of both references [7, 8] where comparison is possible.

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

$$m_i^2 = a^2(\mu_i^2(\Lambda) + \delta m_i^2(\hbar) + \delta m_i^2(\hbar^2)), \tag{51}$$

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

$$\begin{aligned}
\Sigma \approx& 3.17591153625, \quad \zeta \approx 0.08849, \quad \delta \approx 1.942130, \quad \rho \approx -0.313964, \\
\kappa_1 \approx& 0.958382, \quad \kappa_2 = \frac{\Sigma^2}{4} - \frac{\delta}{2} - \frac{1}{4}, \quad \kappa_3 \approx 0.751498, \quad \kappa_4 \approx 1.204295.
\end{aligned} \tag{52}$$

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 (bad notation; here m_ϕ^2 means mass in lattice regularization, with units GeV^2)

$$\langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \rangle_{\text{cont}} = \langle \phi^\dagger \phi \rangle_{\text{cont}} = a^{-1} \langle \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \rangle_{\text{latt}} + \frac{d(\delta V)}{dm_\phi^2}, \tag{53}$$

where

$$\delta V = -2m_\phi^2 \frac{\Sigma}{4\pi a} - \frac{3}{16\pi^2} g^2 m_\phi^2 \Big(\ln \frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta \Big). \tag{54}$$

The full mass counterterm is

$$\begin{aligned}
(\delta m_\phi^2)_{\text{fund}} =& -\Big(\frac{3}{2} g^2 + 6\lambda \Big) \frac{\Sigma}{4\pi a} \\
& - \frac{1}{16\pi^2} \Big[\Big(\frac{51}{16} g^4 + 9\lambda g^2 - 12\lambda^2 \Big) \Big(\ln \frac{6}{a\Lambda} + \zeta \Big) + 9\lambda g^2 \Big(\frac{1}{4}\Sigma^2 - \delta \Big) \\
& + \frac{3}{4} g^4 \Big(\frac{15}{16}\Sigma^2 + \frac{\pi}{3}\Sigma + \frac{5}{4} - \frac{7}{2}\delta - 4\rho + 4\kappa_1 - \kappa_2 - \kappa_3 - 3\kappa_4 \Big) \Big].
\end{aligned} \tag{55}$$

If the hypercharge field is included, the counterterms are instead

$$\delta V_{\text{U(1)}} = -2m_\phi^2 \frac{\Sigma}{4\pi a} - \frac{1}{16\pi^2} (3g^2 + g'^2) m_\phi^2 \Big(\ln \frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta \Big), \tag{56}$$

$$\begin{aligned}
(\delta m_\phi^2)_{\text{fund,U(1)}} =& -\Big(\frac{3}{2} g^2 + \frac{1}{2} g'^2 + 6\lambda \Big) \frac{\Sigma}{4\pi a} \\
& - \frac{1}{16\pi^2} \Big[\Big(\frac{51}{16} g^4 - \frac{9}{8} g^2 g'^2 - \frac{5}{16} g'^4 + 9\lambda g^2 + 3\lambda g'^2 - 12\lambda^2 \Big) \Big(\ln \frac{6}{a\Lambda} + \zeta \Big) \\
& + 3\lambda(3g^2 + g'^2) \Big(\frac{1}{4}\Sigma^2 - \delta \Big) + A g'^4 + B g^2 g'^2 \\
& + \frac{3}{4} g^4 \Big(\frac{15}{16}\Sigma^2 + \frac{\pi}{3}\Sigma + \frac{5}{4} - \frac{7}{2}\delta - 4\rho + 4\kappa_1 - \kappa_2 - \kappa_3 - 3\kappa_4 \Big) \Big].
\end{aligned} \tag{57}$$

These were taken from [10], with some guessing based on similarity of SU(2) and U(1) Feynman diagrams. This was necessary because they only gave numerical values for terms on the last two rows. (See our 2024 paper for exact expressions.)

A.2 SU(2) + adjoint scalar

The triplet condensate, we have:

$$\frac{1}{2}\langle \Sigma^a \Sigma^a \rangle_{\text{cont}} = \langle \mathrm{Tr}\,\Sigma^2 \rangle_{\text{cont}} = a^{-1}\langle \mathrm{Tr}\,\Sigma^2 \rangle_{\text{latt}} + \frac{d(\delta V)}{dm_\Sigma^2}, \tag{58}$$

where the counterterm is the sum of

$$\begin{aligned} \delta V(\hbar) &= -\frac{3}{2} m_\Sigma^2 \frac{\Sigma}{4\pi a} \\ \delta V(\hbar^2) &= -\frac{1}{16\pi^2} 6 g^2 m_\Sigma^2 \Big(\ln\frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta \Big). \end{aligned} \tag{59}$$

The mass counterterm is the sum of

$$(\delta m_\Sigma^2(\hbar))_{\text{adj}} = -(4g^2 + 5b_4)\frac{\Sigma}{4\pi a}, \tag{60}$$

$$\begin{aligned} (\delta m_\Sigma^2(\hbar^2))_{\text{adj}} = &-\frac{1}{16\pi^2}\Big[\Big(20 b_4 g^2 - 10 b_4^2\Big)\Big(\ln\frac{6}{a\Lambda} + \zeta\Big) + 20 b_4 g^2 \Big(\frac{1}{4}\Sigma^2 - \delta\Big) \\ &+ 2g^4\Big(\frac{5}{4}\Sigma^2 + \frac{\pi}{3}\Sigma - 6\delta - 6\rho + 4\kappa_1 - \kappa_2 - \kappa_3 - 3\kappa_4\Big)\Big]. \end{aligned} \tag{61}$$

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. (54) and (59):

$$\delta V = -(2m_\phi^2 + \frac{3}{2}m_\Sigma^2)\frac{\Sigma}{4\pi a} - \frac{3}{16\pi^2} g^2 (m_\phi^2 + 2m_\Sigma^2)\Big(\ln\frac{6}{a\Lambda} + \zeta + \frac{\Sigma^2}{4} - \delta\Big). \tag{62}$$

For the masses, we have

$$m_\phi^2 = a^2(\mu_\phi^2(\Lambda) + (\delta m_\phi^2)_{\text{fund}} + (\delta m_\phi^2)_{\text{both}}) \tag{63}$$

$$m_\Sigma^2 = a^2(\mu_\Sigma^2(\Lambda) + (\delta m_\Sigma^2)_{\text{adj}} + (\delta m_\Sigma^2)_{\text{both}}) \tag{64}$$

where the new contributions read

$$\begin{aligned} (\delta m_\phi^2)_{\text{both}} = &-\frac{3}{2} a_2 \frac{\Sigma}{4\pi a} - \frac{1}{16\pi^2}\Big[\Big(-\frac{3}{4}g^4 + 6a_2 g^2 - \frac{3}{2}a_2^2\Big)\Big(\ln\frac{6}{a\Lambda} + \zeta\Big) \\ &+ 6a_2 g^2\Big(\frac{1}{4}\Sigma^2 - \delta\Big) - 3g^4\rho\Big], \end{aligned} \tag{65}$$

$$\begin{aligned} (\delta m_\Sigma^2)_{\text{both}} = &-2a_2\frac{\Sigma}{4\pi a} - \frac{1}{16\pi^2}\Big[\Big(-g^4 + 3a_2 g^2 - 2a_2^2\Big)\Big(\ln\frac{6}{a\Lambda} + \zeta\Big) \\ &+ 3a_2 g^2\Big(\frac{1}{4}\Sigma^2 - \delta\Big) - 4g^4\rho\Big]. \end{aligned} \tag{66}$$

Note that there are also contributions not proportional to a_2.

TODO U(1) parts!!

A.4 SU(2) + fundamental Higgs + singlet scalar

Add the following terms to the SU(2) + Higgs action:

$$\begin{aligned} S_S =& a \sum_{x,i} \left[S^2 - S(x)S(x+i) \right] \\ &+ a^3 \sum_{x} \Big[b_1 S + \frac{1}{2} m_S^2 S^2 + \frac{1}{3} b_3 S^3 + \frac{1}{4} b_4 S^4 + \frac{1}{2} a_1 S \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi + \frac{1}{2} a_2 S^2 \frac{1}{2} \operatorname{Tr} \Phi^\dagger \Phi \Big]. \end{aligned} \tag{67}$$

Here $m_S^2 = m_S^2(\mu) + \delta m_S^2, b_1 = b_1(\mu) + \delta b_1$ etc with μ being the $\overline{\text{MS}}$ scale.

Neglecting $\mathcal{O}(a)$, the counterterms are:

$$\delta V = -\hbar \frac{\Sigma}{4\pi a} \left(\frac{1}{2} m_S^2 + 2 m_\phi^2 \right) + \frac{1}{(4\pi)^2} \frac{3}{4} g^2 m_\phi^2 \Big[(4\delta - \Sigma^2) - \Big(\ln \frac{6}{a\mu} + \zeta \Big) \Big] + \text{indep of masses and } b_1 \tag{68}$$

$$\delta b_1 = -\hbar \frac{\Sigma}{4\pi a} (b_3 + a_1) + \hbar^2 \frac{1}{(4\pi)^2} \Big[\frac{3}{8} a_1 g^2 (4\delta - \Sigma^2) + \Big(2b_3 b_4 + a_1 (a_2 - \frac{3}{2} g^2) \Big) \Big(\ln \frac{6}{a\mu} + \zeta \Big) \Big] \tag{69}$$

$$\delta m_S^2 = -\hbar \frac{\Sigma}{4\pi a} (2a_2 + 3b_4) + \hbar^2 \frac{1}{(4\pi)^2} \Big[\frac{3}{4} a_2 g^2 (4\delta - \Sigma^2) + \Big(2a_2^2 + 6b_4^2 - 3a_2 g^2 \Big) \Big(\ln \frac{6}{a\mu} + \zeta \Big) \Big] \tag{70}$$

$$\begin{aligned} \delta m_\phi^2 =& -\hbar \frac{\Sigma}{4\pi a} \left(\frac{1}{2} a_2 + \frac{3}{2} g^2 + 6\lambda \right) + \hbar^2 \frac{1}{(4\pi)^2} \Big[\Big(-\frac{51}{16} g^4 - 9 g^2 \lambda + 12 \lambda^2 + \frac{1}{2} a_2^2 \Big) \Big(\ln \frac{6}{a\mu} + \zeta \Big) \\ &+ 9 g^2 \lambda \left(\delta - \frac{1}{4} \Sigma^2 \right) - \frac{3}{4} g^4 \left(\frac{15}{16} \Sigma^2 + \frac{\pi}{3} \Sigma + \frac{5}{4} - \frac{11}{2} \delta - 6\rho + 4\kappa_1 - 2\kappa_4 \right) \Big]. \end{aligned} \tag{71}$$

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.

- Open MPI 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).

- **Update 18.9.2019:** I have crosschecked carefully against David Weir's code using the same seed for drand48(), and verified that our programs produce exactly the same numbers with SU(2) links, Higgs and real triplet all included in the simulation. By "same numbers" I mean that starting from the same initial configuration, fields change exactly the same way in all local updates in both programs, and consequently the measured values for all observables are exactly the same (some care was needed with precision, since David inputs g instead of β_G). The comparison was done by arranging update routines in David's version so that everything happens in the same order as in my implementation. I used serial version of David's code and parallel version of my code (but with 1 MPI node) on a $32 \times 32 \times 32$ lattice. I performed separate tests on full Metropolis and scalar Metropolis + gauge heatbath, and both work (did not compare overrelaxation).

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].

[9] K. Farakos, K. Kajantie, K. Rummukainen and M. E. Shaposhnikov, Phys. Lett. B **336** (1994) 494 doi:10.1016/0370-2693(94)90563-0 [hep-ph/9405234].

[10] K. Kajantie, M. Laine, K. Rummukainen and M. E. Shaposhnikov, Nucl. Phys. B **493** (1997) 413 doi:10.1016/S0550-3213(97)00164-8 [hep-lat/9612006].