`DRalgo`: A package for effective field theory approach for thermal phase transitions ☆,☆☆

Andreas Ekstedt [a,b,c,*], Philipp Schicho [d], Tuomas V.I. Tenkanen [e,f,g]

[a] *Department of Physics and Astronomy, Uppsala University, P.O. Box 256, SE-751 05 Uppsala, Sweden*
[b] *II. Institute of Theoretical Physics, Universität Hamburg, D-22761, Hamburg, Germany*
[c] *Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany*
[d] *Department of Physics and Helsinki Institute of Physics, P.O. Box 64, FI-00014 University of Helsinki, Finland*
[e] *Nordita, KTH Royal Institute of Technology and Stockholm University, Roslagstullsbacken 23, SE-106 91 Stockholm, Sweden*
[f] *Tsung-Dao Lee Institute & School of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai 200240, China*
[g] *Shanghai Key Laboratory for Particle Physics and Cosmology, Key Laboratory for Particle Astrophysics and Cosmology (MOE), Shanghai Jiao Tong University, Shanghai 200240, China*

ARTICLE INFO

ABSTRACT

`DRalgo` is an algorithmic implementation that constructs an effective, dimensionally reduced, high-temperature field theory for generic models. The corresponding Mathematica package automatically performs the matching to next-to-leading order. This includes two-loop thermal corrections to scalar and Debye masses as well as one-loop thermal corrections to couplings. `DRalgo` also allows for integrating out additional heavy scalars. Along the way, the package provides leading-order beta functions for general gauge-charges and fermion-families; both in the fundamental and in the effective theory. Finally, the package computes the finite-temperature effective potential within the effective theory. The article explains the theory of the underlying algorithm while introducing the software on a pedagogical level.

Program summary

References
[1] R. M. Fonseca, Comput. Phys. Commun. 267 (2021) 108085, arXiv:2011.01764.
[2] S. P. Martin, Phys. Rev. D 96 (2017) 096005, arXiv:1709.02397.
[3] S. P. Martin and H. H. Patel, Phys. Rev. D 98 (2018) 076008, arXiv:1808.07615.
[4] M. E. Machacek and M. T. Vaughn, Nucl. Phys. B 249 (1985) 70.
[5] M. E. Machacek and M. T. Vaughn, Nucl. Phys. B 236 (1984) 221.
[6] M. E. Machacek and M. T. Vaughn, Nucl. Phys. B 222 (1983) 83.

☆ The review of this paper was arranged by Prof. Z. Was.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
* Corresponding author.
E-mail addresses: andreas.ekstedt@desy.de (A. Ekstedt), philipp.schicho@helsinki.fi (P. Schicho), tuomas.tenkanen@su.se (T.V.I. Tenkanen).

Contents

1. Introduction

The origin of Baryon asymmetry [1] in the universe remains obscure. As such, much powder has been spent throughout past decades to find a sound explanation of this asymmetry [2–5]. Amongst the suggested mechanisms, the one based on the electroweak phase transition – electroweak baryogenesis – stands out. In this mechanism, generating a baryon asymmetry requires a first-order phase transition, which proceeds by nucleating true-vacuum bubbles. This provides a departure from thermal equilibrium, which is necessary to generate a baryon asymmetry [6]. While the Standard Model (SM) has no strong first-order transition on its own [7–9], its extensions can contain myriads of them. For new field content to trigger a strong first-order phase transition, their masses have to be around the electroweak scale, and their interaction with the SM Higgs cannot be too feeble. Beyond-the-Standard-Model (BSM) theories that exhibit such transitions provide a direct target for many future-generation colliders [10,11]. Furthermore, thermal phase transitions can, perchance, generate gravitational waves that are observable by next-generation space-based detectors such as LISA [12], DECIGO [13], Taiji [14] and BBO [15]. These waves might open a fresh window into the early universe – and the underlying quantum field theory.

The interplay between BSM phenomenology and gravitational waves is among the most actively studied topics in the high-energy-physics literature [16–26]. This interplay requires solid understanding of thermodynamic properties of different models. It has been long known that determining thermodynamics in theories with non-Abelian gauge fields is challenging because of the Linde problem [27]. In short, there are non-perturbative effects arising from massless vector-bosons: these infrared (IR) effects can only be captured by lattice simulations [28]. Despite this, leading-order perturbation theory is frequently used as an approximation. However, it has been pointed out that such leading-order studies contain large theoretical uncertainties, due to slow convergence of perturbation theory [21,29–31].

Dimensional reduction [32,33] offers a way to overcome these challenges. In this framework, ultraviolet (UV) modes – related to non-zero Matsubara modes in the imaginary time formalism – are integrated out. The resulting effective field theory (EFT) [34,35] describes IR, or long wavelength zero modes, living in three spatial dimensions (3d) (cf. also [36–38]). The 3d EFT can be simulated on a lattice and hence by-pass the Linde problem [28,39]. Dimensional reduction can also be viewed as a systematic scheme for thermal resummations, used purely within the realm of perturbation theory [30]. To leading order, dimensional reduction is equivalent to resummation via thermal masses. Concretely, the effective potential, that describes the free energy of thermal plasma, is commonly given in a schematic form [29]

$$V_{\text{eff}} \simeq V_{T=0} + V_T + V_{\text{daisy}} \,. \tag{1.1}$$

Here, $V_{T=0}$ is the Coleman-Weinberg zero-temperature contribution, which is often augmented with the effective potential at one-loop level. Namely, V_T is the thermal correction at one-loop (commonly in the high-temperature expansion) and

$$V_{\text{daisy}} \simeq -\frac{T}{12\pi}\Big[(m^2 + \Pi_T)^{\frac{3}{2}} - (m^2)^{\frac{3}{2}}\Big] \,. \tag{1.2}$$

Here m^2 represents a field-dependent mass and $\Pi_T \sim g^2 T^2$ is a one-loop thermal correction. It is the first term in eq. (1.2) that implements the resummation.[1] This resummed mass, for zero Matsubara modes arises from thermal corrections due to non-zero Matsubara modes. Physically, this corresponds to thermal screening: UV modes in the plasma screen IR modes. In eq. (1.1) all contributions are evaluated at one-loop order, which means that only masses are resummed. However, at high temperatures, the perturbative expansion is controlled by g, rather than g^2 as in a zero-temperature loop expansion. Dimensional reduction makes it possible to systematically include higher order resummations, in particular two-loop thermal masses and one-loop resummation of couplings and fields which are also affected by thermal screening. Hence, dimensional reduction can be used to include important next-to-leading order thermal effects. In particular, renormalization-group (RG) improvements require two-loop computations at high temperatures [29,31]; these are straightforward to implement in the 3d EFT approach [30].

To date there are sundry studies using dimensional reduction in electroweak theories [21,22,24,26,30,34,36–38,40–43]. For similar studies in hot QCD, see [44,45]. Still, many models remain to be studied within the dimensional-reduction framework [46–49].

Along with this article, we launch the Mathematica package `DRalgo` to implement dimensional reduction automatically for user-defined models. This package calculates parameters in the 3d EFT. For example, `DRalgo` calculates two-loop thermal masses, effective couplings, and beta functions. The package also allows users to integrate out heavy scalars and temporal (Debye) scalars. Furthermore, `DRalgo` can be used to compute the two-loop thermal effective potential within the effective theory. Subsequently, the 3d EFT matching relations can be implemented to lattice Monte Carlo simulations. The Algorithm 1 illustrates this pipeline. The goal of this pipeline is to compute thermodynamic properties of phase transitions, such as the critical temperature and the latent heat.[2] This package can be applied

---

Algorithm 1 `DRalgo` algorithm outline in alignment with fig. 2 of [50]. The use of lattice resources is indicated as an optional path. The functions `PerformDRhard[]`, `PerformDRsoft[]` and `CalculatePotentialUS[]` are part of `DRalgo`.

Input: Four-dimensional Lagrangian $\mathcal{L}_{4d}$ with parameters $\{c_1, \ldots, c_n\}$, temperature T, physical parameters, heavy masses $\boldsymbol{M}$
Start: Initialize model
Call `PerformDRhard[]` {
    for all $c_i \in \{c_1, \ldots, c_n\}$ **do**
        Compute 4-dimensional β-functions $\beta(c_i)$
        Compute $c_{i,3d}(T, \boldsymbol{M})$ by integrating out non-zero Matsubara modes
    end for
    Compute thermal (Debye) masses $m_{D,i}(T, \boldsymbol{M})$
    Compute couplings that involve temporal vector fields
}
Output/Input: Three-dimensional soft Lagrangian $\mathcal{L}_{3d}$ with parameters $\{c_{1,3d}, \ldots, c_{n,3d}\}$
Call `PerformDRsoft[]` {
    for all $c_{i,3d} \in \{c_{1,3d}, \ldots, c_{n,3d}\}$ **do**
        Compute 3-dimensional β-functions $\beta_{3d}(c_{i,3d})$
        Compute $\bar{c}_{i,3d}(T, c_{1,3d}, \ldots, c_{n,3d})$ by integrating out massive temporal scalars
    end for
}
Output: Three-dimensional ultrasoft Lagrangian $\bar{\mathcal{L}}_{3d}$ with parameters $\{\bar{c}_{1,3d}, \ldots, \bar{c}_{n,3d}\}$
if Lattice resources **then**
    Compute lattice continuum relations to construct $\mathcal{L}_{3d}^{lattice}$
    Monte Carlo simulation
else
    Call `CalculatePotentialUS[]` {
        Compute effective potential $V_{eff}^{3d}(m_{i,3d}^2)$ up to two-loops
    }
end if
Compute thermodynamic parameters T_c, L/T_c^4

---

to models previously studied in the electroweak and dark-sector phase-transition literature.

Dimensional reduction can also be used to study dynamical systems close to equilibrium. In particular, systems that evolve on a characteristic time scale $t \sim (g^4 T)^{-1}$ can often be described by a Langevin equation [51–53]; in this case the dimensionally reduced potential appears in the equations of motion.

The remainder of this article is organised as follows. Section 1.1 briefly introduces dimensional reduction and matching relations. Section 3 explicates the front-end of the package, its installation, and gives a tutorial based on the Abelian-Higgs model. Section 2 reviews theory and computations in the back-end of the package. Section 4 illustrates additional features by implementing a two-Higgs doublet model. In section 6, we discuss future prospects and possible updates of the package. Finally, appendix A displays the computation of thermodynamical observables using the output of `DRalgo` for the Abelian-Higgs model.

1.1. Prologue: dimensional reduction at next-to-leading-order

At leading order (LO) scalars obtain familiar one-loop thermal masses, while temporal (longitudinal) scalars get Debye masses. At next-to-leading order (NLO) couplings are resummed, and scalars receive two-loop thermal masses. `DRalgo` also calculates two-loop Debye masses. Corrections at NLO are particularly important due to large logarithms. The idea with dimensional reduction is to render some

---

[1] The second $\sim Tm^3$ term in eq. (1.2) removes double counting; the same contribution is included V_T but with unresummed mass. This term is related to the Matsubara zero-mode contribution.

[2] The critical temperature, T_c, is defined by $\Delta V_{eff}^{3d}(T_c) = 0$, where $\Delta V_{eff}^{3d}(T_c)$ denotes the energy-difference between two phases. The strength of the transition depends on the latent heat, L, which is given by $L = T \Delta V_{eff}^{3d} + T^2 \Delta \frac{dV_{eff}^{3d}}{dT}$, where $\Delta \equiv (\ldots)_{high-T} - (\ldots)_{low-T}$ denotes the difference between the high- and low-temperature phase.

logarithms ($\ln \frac{\bar{\mu}}{T}$) small by matching at a high energy ($\bar{\mu} \sim T$). The remaining logarithms can then be resummed by RG-evolution within the EFT.

Phase transitions often occur below the Debye-mass scale. In such cases temporal scalars can also be integrated out. Thus effectively removing large logarithms of the form $\ln \frac{m_D}{\mu}$, where m_D is a Debye mass. This second EFT is said to live at the ultrasoft scale, which is characterised by energies of $\mathcal{O}(g^2 T)$. In summary,[3] the hard scale corresponds to energies $E \sim T$; the soft scale corresponds to energies $E \sim gT$; and the ultrasoft scale corresponds to energies $E \sim g^2 T$.

To illustrate next-to-leading order dimensional reduction, we consider a schematic model with scalar mass parameter μ^2, scalar quartic coupling λ, and gauge coupling g. Given the power counting $\mu^2 \sim g^2 T^2$, $\lambda \sim g^2$, the matching of the mass parameter is

$$
\bar{\mu}_3^2 = \underbrace{\boxed{\overset{\text{tree-level}}{\mu^2} + \overset{\text{1-loop}}{\#g^2 T^2}}}_{\mathcal{O}(g^2)} + \underbrace{\boxed{\overset{\text{1-loop}}{\#g^2 \mu^2} + \overset{\text{2-loop}}{\#g^4 T^2}}}_{\mathcal{O}(g^4)} + \mathcal{O}(g^6)
$$

$$
+ \underbrace{\boxed{\overset{\text{1-loop}}{\#g^2 m_D}}}_{\mathcal{O}(g^3)} + \underbrace{\boxed{\overset{\text{2-loop}}{\#g^4}}}_{\mathcal{O}(g^4)} + \mathcal{O}(g^5) ,
\tag{1.3}
$$

where the first line (with even powers of g) results from the first step, and the second line (with odd power of g) from second step of the dimensional reduction. In practice, *full* $\mathcal{O}(g^4)$ contributions are included. Going to higher orders, requires a three-loop computation for both steps of the dimensional reduction. The situation is similar for the coupling:

$$
\bar{\lambda}_3 = \underbrace{\boxed{\overset{\text{tree-level}}{T\lambda}}}_{\mathcal{O}(g^2)} + \underbrace{\boxed{\overset{\text{1-loop}}{\#g^4}}}_{\mathcal{O}(g^4)} + \mathcal{O}(g^6)
$$

$$
+ \underbrace{\boxed{\overset{\text{1-loop}}{\#\frac{g^4}{m_D}}}}_{\mathcal{O}(g^3)} + \underbrace{\boxed{\overset{\text{2-loop}}{\#\frac{g^6}{m_D^2}}}}_{\mathcal{O}(g^4)} + \mathcal{O}(g^5) .
\tag{1.4}
$$

In practice for the coupling, $\mathcal{O}(g^4)$ pieces are neglected since their numerical effect is small (despite being formally of the same order). These contributions arise during the second step of the dimensional reduction at two-loop. Pursuing higher order, requires a two-loop computation for the first step, and a three-loop computation for the second step of the dimensional reduction.

In perturbation theory the definition of 3d EFT parameters is accompanied with a perturbative computation of the effective potential [30]:

$$
V_{\text{eff}}^{\text{3d}} = \underbrace{V_{\text{tree}}^{\text{3d}}}_{\mathcal{O}(g^2)} + \underbrace{V_{\text{1-loop}}^{\text{3d}}}_{\mathcal{O}(g^3)} + \underbrace{V_{\text{2-loop}}^{\text{3d}}}_{\mathcal{O}(g^4)} + \mathcal{O}(g^5) .
\tag{1.5}
$$

Instead of expanding the result in terms of 4d parameters of the parent theory, resummed couplings and masses are kept along. This improves the overall convergence as the result is less sensitive to the renormalization scale [54,55]. The order $\mathcal{O}(g^5)$ requires a computation at three-loop level [56–58].

Dimensional reduction from a zero-temperature EFT perspective

It is instructive to consider the same physics from a standard EFT perspective. In particular, for equilibrium observables we can, analogous to Kaluza-Klein theories [59], view thermal corrections as an infinite tower of heavy particles. To see the connection with dimensional reduction, we consider a theory with one light and one heavy scalar in Euclidean spacetime:

$$
\mathcal{L} = \frac{1}{2}(\partial_\mu \phi)^2 + \frac{1}{2}(\partial_\mu \Phi)^2 + \frac{1}{2}m^2 \phi^2 + \frac{1}{2}M^2 \Phi^2 + \frac{1}{4}\kappa \phi^2 \Phi^2 + \frac{1}{4!}\lambda \phi^4 .
\tag{1.6}
$$

Imagine now that there exists a hierarchy $M^2 \gg m^2$. This is a well-known situation, and when calculating scattering processes one encounters large corrections to m^2 scaling as κM^2 – analogous to the hierarchy problem. As well as large logarithms in the form $\ln \frac{\bar{\mu}^2}{M^2}$ or $\ln \frac{\bar{\mu}^2}{m^2}$. One of these logarithms will then be large regardless the choice of the RG-scale $\bar{\mu}$.

Using thermal masses is equivalent to using a resummation[4] $m^2 \to m^2 + a\kappa M^2$. This resummation does, however, not solve our problems of large logarithms. Therefore, calculations, even with thermal masses, are sensitive to the RG-scale. The situation is much improved by using an EFT. Therein, the logarithms $\ln \frac{\bar{\mu}^2}{M^2}$ are rendered small by matching at the scale $\bar{\mu}_{\text{Match}} \sim M$, and the remaining logarithms $\ln \frac{\bar{\mu}^2}{m^2}$ are taken care of by RG-evolution in the low-energy theory [60].

Once the scalar field Φ has been integrated out, the resulting EFT is of the form

$$
\mathcal{L}_{\text{eff}} = \frac{1}{2}(\partial_\mu \phi)^2 + \frac{1}{2}m_{\text{eff}}^2 \phi^2 + \frac{1}{4!}\lambda_{\text{eff}} \phi^4 .
\tag{1.7}
$$

---

[3] Some literature [34] interchangeably refers to the hard scale as superheavy, the soft scale as heavy, and the ultrasoft scale as light.

[4] This can also be viewed as a temperature-dependent renormalization $\delta_T m^2$ such that $m^2 = m^2 + \delta_T m^2$ [24].

To leading order $m_{\text{eff}}^2 = m^2 + a\kappa M^2$, and $\lambda_{\text{eff}} = \lambda$. While at NLO

$$\delta m_{\text{eff}}^2 = \kappa^2 M^2 \left(b + c \ln \frac{\bar{\mu}_{\text{Match}}^2}{M^2} \right) , \tag{1.8}$$

$$\delta \lambda_{\text{eff}} = \kappa^2 \left(d + e \ln \frac{\bar{\mu}_{\text{Match}}^2}{M^2} \right) , \tag{1.9}$$

where b, c, d, e are numerical coefficients. It is precisely these kind of corrections that appear in the dimensionally reduced theory.

Finally, to do calculations in the IR, we should run the couplings within the EFT from $\bar{\mu}_{\text{Match}} \sim M$ down to $\bar{\mu}_{\text{IR}} \sim m_{\text{eff}}$. This way, all large logarithms are eliminated and perturbative convergence is improved.

When studying equilibrium phenomena it is common to use the imaginary-time formalism. Because fermions and bosons are (anti-)periodic functions with respect to imaginary time in this formalism, it is possible to expand the fields in a fourier series [61]

$$\phi(\tau, x) = T \sum_{n=-\infty}^{\infty} \phi_n(x) e^{i\omega_n \tau} , \quad \omega_n = \begin{cases} \omega_n^B = 2n\pi T & \text{(bosons)} \\ \omega_n^F = (2n+1)\pi T & \text{(fermions)} \end{cases} , \quad n \in \mathbb{Z} , \tag{1.10}$$

where ω_n^B and ω_n^F are known as Matsubara modes. So at large temperatures we effectively have an infinite tower of heavy particles with masses $M_n^2 = \left[(2n+\sigma)\pi T \right]^2$ for all integers $n \neq 0$, and $\sigma = 0(1)$ for bosons(fermions). Hence, the high-temperature matching scale is $\mu_{\text{Match}} \sim T$ and the low-energy scale is $\mu_{\text{IR}}^2 \sim g^2 T^2$. Since we are describing equilibrium (static) processes, there is no time dependence – the degrees of freedom of the EFT are static and live in three dimensions.

2. Theory behind the scenes: dimensional reduction of a generic model

In this section, we dig deeper into the theory and computations in the back-end of the software. The implementation for dimensional reduction of a generic model is based on tensor-notation [62–66] that separates Lorentz algebra from group algebra. The Lorentz algebra is hard-coded, with contractions done by `FeynCalc` [67,68] and `FORM` [21,69], while the group-structure is provided by the user.

2.1. Lagrangian in the 4d fundamental theory

Consider a general Lagrangian in Minkowski spacetime with mostly-plus metric: $g^{\mu\nu} = \text{diag}(-1, 1, 1, 1)$, $\{\sigma_\mu, \overline{\sigma}_\nu\} = -2g_{\mu\nu}$. In terms of the functional-integral measure and the action S, the partition function is $\mathcal{Z} = \int \mathcal{D}e^{iS}$. Correspondingly, the sigma matrices are defined as

$$\sigma^\mu = \left(\mathbb{1}, \sigma^i \right), \quad \overline{\sigma}^\mu = \left(-\mathbb{1}, \sigma^i \right) , \tag{2.1}$$

where σ^i are Pauli matrices.

To write down the Lagrangian, in a flattened form, we employ a basis where all scalars and vectors are real. In addition, all fermions are composed of two-component Weyl-spinors [70]. The most general, four dimensional, Lagrangian in Minkowski space is [35,62,63]

$$\mathcal{L} = -\frac{1}{2} R_i (-\delta_{ij}\partial_\mu \partial^\mu + \mu_{ij}) R_j - \frac{1}{4} F_{\mu\nu}^a F^{\mu\nu,a} \delta_{ab} - \frac{1}{2\xi_a} (\partial_\mu A^{a,\mu})^2$$
$$- \partial^\mu \overline{\eta}^a \partial_\mu \eta^a + i\psi_I^\dagger \overline{\sigma}^\mu \partial_\mu \psi^I - \frac{1}{2} (M^{IJ}\psi_I \psi_J + \text{h.c.}) + \mathcal{L}_{\text{int}} , \tag{2.2}$$

$$\mathcal{L}_{\text{int}} = -\lambda_i R_i - \frac{1}{3!} \lambda_{ijk} R_i R_j R_k - \frac{1}{4!} \lambda_{ijkm} R_i R_j R_k R_m - \frac{1}{2}(Y^{iIJ} R_i \psi_I \psi_J + h.c)$$
$$+ g_{IJ}^a A_\mu^a \psi_I^\dagger \overline{\sigma}^\mu \psi_J - g_{jk}^a A_\mu^a R_j \partial^\mu R_k - \frac{1}{2} g_{jn}^a g_{kn}^b A_\mu^a A^{\mu,b} R_j R_k - g^{abc} A^{\mu,a} A^{\nu,b} \partial_\mu A_\nu^c$$
$$- \frac{1}{4} g^{abe} g^{cde} A^{\mu a} A^{\nu b} A_\mu^c A_\nu^d + g^{abc} A_\mu^a \eta^b \partial^\mu \overline{\eta}^c , \tag{2.3}$$

where $\mathcal{L}_{\text{int}}$ is the interaction Lagrangian and $F_{\mu\nu} = \frac{i}{g}[D_\mu, D_\nu]$ is the field strength tensor with the corresponding gauge coupling g. In this notation, the field R_i denotes a real scalar-field with scalar-index i. In the Standard Model, i corresponds to the Higgs, neutral Goldstone, and real/imaginary component of the charged Goldstone. Further, the field A_μ^a corresponds to a real vector-field with vector-index a, the field η is a ghost-field and the field ψ_I is a Weyl-spinor with fermion-index I. In addition, μ_{ij} denote (squared) scalar masses, and M_{IJ} fermion masses. Repeated indices are always summed over irrespective of their vertical placement.

Above we denoted Y_{iIJ} as Yukawa couplings, λ_{ijkl} as scalar quartic, λ_{ijk} as scalar cubic, and λ_i as scalar tadpole couplings. The gauge couplings g_{jk}^a are (anti-symmetric) representation matrices. For example, for an adjoint scalar-representation in SO(3) $g_{jk}^a = \epsilon^{ajk}$. The abbreviation h.c. stands for the hermitian conjugate. The vertical placement of fermion indices is important; e.g. $M^{IJ} = M_{IJ}^*$. We refer to [70] for further details.

In the dimensional-reduction step, hard modes with masses $\sim \pi T$ are integrated out. This is done by matching the fundamental Lagrangian above to the effective Lagrangian living in three-dimensions. To avoid large logarithms this matching should be performed close to $\bar{\mu} = \pi T$ where $\bar{\mu}$ is the RG-scale. This three-dimensional Lagrangian does not contain fermions. Moreover, in the dimensional-reduction step the temporal component of vectors – represented by temporal scalar fields in the EFT – obtain Debye masses, as well as thermally generated interactions with other scalars.

The matching is straightforward using Euclidean signature. This is achieved by redefining

$$A_0^{M,a} \equiv i A_0^{E,a}, \quad \sigma_i^M = -i\sigma_i^E, \quad t = -i\tau, \quad \partial_0^M = i\partial_0^E, \tag{2.4}$$

$$(\partial_\mu R \, \partial^\mu R)_M = (\partial_\mu R \, \partial^\mu R)_E, \quad \slashed{\partial}_M = i\slashed{\partial}_E, \quad \{\sigma_\mu, \overline{\sigma}_\nu\}_E = 2\delta_{\mu\nu}, \tag{2.5}$$

where quantities are denoted either in Minkowskian (M) or Euclidean (E) metric. Henceforth, we suppress these subscripts and implicitly assume an Euclidean metric.

With the above redefinitions, the partition function is defined as $\mathcal{Z}_E = \int \mathcal{D} e^{-S_E}$ with the most general tree-level Lagrangian in Euclidean signature

$$\mathcal{L} = \frac{1}{2} R_i (-\delta_{ij}\partial_\mu \partial^\mu + \mu_{ij}) R_j + \frac{1}{4} F_{\mu\nu}^a F^{\mu\nu,a}\delta_{ab} + \frac{1}{2\xi_a}(\partial_\mu A^{a,\mu})^2$$

$$+ \partial^\mu \overline{\eta}^a \partial_\mu \eta^a + \psi_I^\dagger \overline{\sigma}^\mu \partial_\mu \psi^I + \frac{1}{2}(M^{IJ}\psi_I\psi_J + \text{h.c.}) + \mathcal{L}_{\text{int}}, \tag{2.6}$$

$$\mathcal{L}_{\text{int}} = \lambda_i R_i + \frac{1}{3!}\lambda_{ijk} R_i R_j R_k + \frac{1}{4!}\lambda_{ijkm} R_i R_j R_k R_m + \frac{1}{2}(Y^{iIJ} R_i \psi_I \psi_J + \text{h.c.})$$

$$+ i g_{IJ}^a A_\mu^a \psi_I^\dagger \overline{\sigma}^\mu \psi_J + g_{jk}^a A_\mu^a R_j \partial^\mu R_k + \frac{1}{2} g_{jn}^a g_{kn}^b A_\mu^a A^{\mu,b} R_j R_k + g^{abc} A^{\mu,a} A^{\nu,b} \partial_\mu A_\nu^c$$

$$+ \frac{1}{4} g^{abe} g^{cde} A^{\mu a} A^{\nu b} A_\mu^c A_\nu^d - g^{abc} A_\mu^a \eta^b \partial^\mu \overline{\eta}^c. \tag{2.7}$$

All Lorentz indices are contracted using the Euclidean metric $\delta_{\mu\nu} = \text{diag}(+1,+1,+1,+1)$. The Feynman rules for the vertices are

$$R_i R_j R_k R_l : -\lambda_{ijkl}, \tag{2.8}$$

$$R_i R_j R_k : -\lambda_{ijk}, \tag{2.9}$$

$$R_i : -\lambda_i, \tag{2.10}$$

$$R_i \psi_I \psi_J : -Y_{iIJ}, \tag{2.11}$$

$$\psi_I \psi_J A_\mu^a : -i g_{IJ}^a \overline{\sigma}_\mu, \tag{2.12}$$

$$R_i R_j A_\mu^a : i g_{ij}^a (p+q)_\mu, \tag{2.13}$$

$$A_\mu^a A_\nu^b A_\rho^c : i g^{abc} T_{\mu\nu\rho}, \tag{2.14}$$

$$A_\mu^a A_\nu^b A_\rho^c A_\sigma^d : -G_{\mu\nu\rho\sigma}^{abcd}, \tag{2.15}$$

$$\overline{\eta}^a \eta^b A_\mu^c : i g^{abc} p_\mu, \tag{2.16}$$

$$R_i R_j A_\mu^a A_\nu^b : \delta_{\mu\nu}\left(g_{in}^a g_{nj}^b + g_{jn}^a g_{ni}^b\right) \equiv \delta_{\mu\nu} H_{ij}^{ab}, \tag{2.17}$$

where $G_{\mu\nu\rho\sigma}^{abcd}$ and $T_{\mu\nu\rho}$ are defined in [63]. Since we only focus on the matching, only the symmetric phase is relevant and vacuum expectation values are not introduced. The propagators are

$$\langle R_i(p) R_j(q) \rangle = \frac{\delta_{ij}\delta(p+q)}{p^2},$$

$$\langle A_\mu^a(p) A_\nu^b(q) \rangle = \frac{\delta^{ab}\delta(p+q)}{p^2} P_{\mu\nu}(p),$$

$$\langle \overline{\eta}^a(p) \eta^b(q) \rangle = \frac{\delta^{ab}\delta(p-q)}{p^2},$$

$$\langle \psi_I(p) \psi_J(q) \rangle = \frac{\delta_{IJ}\delta(p-q) i p_\mu \sigma_\mu}{p^2}, \tag{2.18}$$

where all momenta are incoming by convention and

$$P_{\mu\nu}(p) = \delta_{\mu\nu} - (1-\xi)\frac{p_\mu p_\nu}{p^2}, \tag{2.19}$$

where ξ is the corresponding gauge parameter which for our investigations in Landau gauge is set to $\xi = 0$. All the Lorentz structure is contained in the vertices, and the generalized coupling constants take care of the group structure.

To do the matching we need to renormalize our theory. The matching can, and will, introduce kinetic mixing. To allow for this, we express the bare (b) fields and scalar masses as

$$R_{i(b)} = Z_{ij}^{1/2} R_j, \quad \mu_{ij(b)} = Z_{ij}^\mu = \mu_{ij} + \delta\mu_{ij}, \quad A_{\mu(b)}^a = Z_{ab}^{1/2} A_\mu^b, \quad \psi_{I(b)} = Z_{IJ}^{1/2}\psi_J. \tag{2.20}$$

By construction all propagators should be diagonal at tree-level, to wit

$$Z_{ik}^{1/2} Z_{kj}^{1/2} = \delta_{ij} + \delta Z_{ij}, \quad Z_{ac}^{1/2} Z_{cb}^{1/2} = \delta^{ab} + \delta Z^{ab}. \tag{2.21}$$
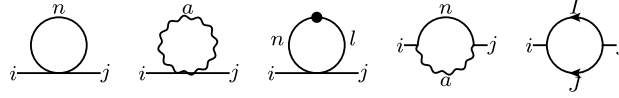
Fig. 1. Diagrams contributing to the thermal scalar masses. A wiggly line denotes gauge bosons, a black line scalar particles, and a directed black line fermions. A black dot represents corresponding counterterms.

Assumed size of masses and couplings

For the current problem there are three energy scales: πT, gT, and $g^2 T$. Denoted as the hard, soft, and ultrasoft scale, respectively. In `DRalgo` it is assumed that all scalar (fermion) masses scale as gT or $g^2 T$. If, on the contrary, some masses would be hard, then they should formally be integrated out together with high-temperature modes.[5] After hard modes are integrated out, it is up to the user whether soft-scale (gT) fields are integrated out as well. By default all temporal scalars have masses of this order, but some models might also have additional scalars with soft masses. For the couplings, `DRalgo` assumes that all quartics and cubics scale as g^2. This scaling is chosen so that scalar masses of $\mathcal{O}(gT)$ can be integrated out safely [24].

In summary, `DRalgo` assumes the following scaling

$$\lambda_{ijkl} \sim g^2 , \qquad \lambda_{ijk} \sim g^2 , \qquad \lambda_i \sim g^2 ,$$
$$\mu_{ij} \sim g^2 T^2 , \qquad M_{IJ} \sim gT , \qquad Y_{iIJ} \sim g , \qquad g^a_{IJ} \sim g^a_{ij} \sim g^{abc} \sim g . \tag{2.22}$$

It is of course possible to consider smaller couplings and masses than those above.

2.2. Lagrangian in the 3d effective theory

The three-dimensional theory is of the form

$$\mathcal{L} = \frac{1}{2} R_i (-\delta_{ij} \partial_\mu \partial^\mu + \mu_{ij}) R_j + \frac{1}{4} F^a_{\mu\nu} F^{\mu\nu,a} \delta_{ab} + \frac{1}{2\xi_a} (\partial_\mu A^{a,\mu})^2$$
$$+ \partial^\mu \overline{\eta}^a \partial_\mu \eta^a + \frac{1}{2} \partial_\mu A^a_0 \partial^\mu A^a_0 + \frac{1}{2} \mu^{ab}_D A^a_0 A^b_0 + \mathcal{L}_{\text{int}} , \tag{2.23}$$

$$\mathcal{L}_{\text{int}} = h_i R_i + \frac{1}{3!} h_{ijk} R_i R_j R_k + \frac{1}{4!} h_{ijkm} R_i R_j R_k R_m$$
$$+ g^a_{jk} A^a_\mu R_j \partial^\mu R_k + \frac{1}{2} g^a_{jn} g^b_{kn} A^a_\mu A^{\mu,b} R_j R_k + g^{abc} A^{\mu,a} A^{\nu,b} \partial_\mu A^c_\nu$$
$$+ \frac{1}{4} g^{abe} g^{cde} A^{\mu a} A^{\nu b} A^c_\mu A^d_\nu - g^{abc} A^a_\mu \eta^b \partial^\mu \overline{\eta}^c + \frac{1}{2} g^{ace} g^{bed} A^a_\mu A^{\mu,b} A^c_0 A^d_0$$
$$+ \frac{1}{4!} \lambda^{abcd}_A A^a_0 A^b_0 A^c_0 A^d_0 + \frac{1}{4} g0^{ab}_{ij} A^a_0 A^b_0 R_i R_j + g^{abc}_0 A^a_\mu A^b_0 \partial^\mu A^c_0 , \tag{2.24}$$

where Lorentz indices range between $\mu, \nu = \{1, \ldots, 3\}$. In renormalizing the fields, we define

$$Z^{1/2}_{ik} Z^{1/2}_{kj} = \delta_{ij} + \delta Z_{3,ij} , \quad Z^{1/2}_{ac} Z^{1/2}_{cb} = \delta^{ab} + \delta Z^{ab}_3 , \quad Z^{1/2}_{ac,L} Z^{1/2}_{cb,L} = \delta^{ab} + \delta Z^{ab,L}_3 . \tag{2.25}$$

Here, the term $\delta Z^{ab,L}_3$ corresponds to temporal vectors. The propagators of this theory are as in eq. (2.18) in three dimensions with the absence of the fermionic propagator and the inclusion of the temporal-vector propagator

$$\langle A^a_0(p) A^b_0(q) \rangle = \frac{\delta^{ab} \delta(p+q)}{p^2} . \tag{2.26}$$

The counterterm Feynman rules are the same as above, barring the new temporal-vector diagram

$$A^a_0 A^b_0 : -p^2 \delta Z^{ab,L}_3 - \delta \mu^{ab}_D . \tag{2.27}$$

2.3. Matching

Self-energies and general n-point correlations are calculated both in the original 4d theory and the effective 3d theory. The 3-dimensional parameters are chosen, or matched, to give the same correlators as the original theory [30,34].

One-loop thermal scalar masses

We start with the matching for scalar masses. For the matching we assume to be in a regime of soft or ultrasoft external momenta $p \sim gT$ or $p \sim g^2 T$. The corresponding diagrams at one-loop level are illustrated in Fig. 1. In their computation, we need the master integrals [71]

---

5 This will be implemented in future versions of `DRalgo`.

$$I_\alpha^{4b} = \sum_Q \frac{1}{[Q^2]^\alpha} = \left(\frac{\bar{\mu}^2 e^{\gamma_E}}{4\pi}\right)^\epsilon 2T \frac{[2\pi T]^{d-2\alpha}}{(4\pi)^{\frac{d}{2}}} \frac{\Gamma\left(\alpha - \frac{d}{2}\right)}{\Gamma(\alpha)} \zeta_{2\alpha - d} , \tag{2.28}$$

$$I_\alpha^{4f} = \sum_{\{Q\}} \frac{1}{[Q^2]^\alpha} = \left(\frac{\bar{\mu}^2 e^{\gamma_E}}{4\pi}\right)^\epsilon 2T \frac{[2\pi T]^{d-2\alpha}}{(4\pi)^{\frac{d}{2}}} \frac{\Gamma\left(\alpha - \frac{d}{2}\right)}{\Gamma(\alpha)} \left(1 - 2^{d-2\alpha}\right) \zeta_{2\alpha - d} , \tag{2.29}$$

where $\zeta_s = \zeta(s)$ is the Riemann zeta function. The d-dimensional bosonic integral measure is defined as

$$\sum_Q \equiv T \sum_{q_n} \left(\frac{\bar{\mu}^2 e^{\gamma_E}}{4\pi}\right)^\epsilon \int \frac{d^d \mathbf{q}}{(2\pi)^d} , \tag{2.30}$$

while for fermions the summation is written as $\sum_{\{Q\}}$. We set $d = 3 - 2\epsilon$ with Euclidean four-momenta $Q^2 = q_n^2 + \mathbf{q}^2 = \left[(2n+\sigma)\pi T\right]^2 + \mathbf{q}^2$, and $\sigma = 0(1)$ for bosons(fermions).

The first diagram in Fig. 1 gives

$$D_1^{ij} = -\frac{1}{2}\lambda_{ijnn} I_1^{4b} , \tag{2.31}$$

where the factor of $\frac{1}{2}$ is the symmetry factor, $-\lambda_{ijnn}$ is the Feynman rule, and

$$I_1^{4b} = \frac{T^2}{12} + \mathcal{O}(\epsilon) . \tag{2.32}$$

The zero Matsubara mode is here ignored. The reason for this is that the zero Matsubara mode corresponds to soft momenta, and should not be integrated out. Conversely, since the $n = 0$ mode exists both in the EFT and the parent theory, this contribution cancels in the matching. The second diagram is likewise (neglecting higher ϵ terms)

$$D_2^{ij} = \frac{1}{2}H_{ij}^{aa} dI_1^{4b} , \quad dI_1^{4b} = \frac{T^2}{4} , \tag{2.33}$$

where we used the shorthand notation for H_{ij}^{ab} from eq. (2.17).

The remaining diagrams give

$$D_3^{ij} = i^2 g_{in}^a g_{nj}^a \frac{3}{(4\pi)^2 \epsilon_b} p^2 , \quad D_4^{ij} = \frac{1}{2}\lambda_{ijnl}\mu_{nl} \frac{1}{(4\pi)^2 \epsilon_b} . \tag{2.34}$$

The first diagram contains the external momentum is p, and the last diagram is the contribution from a scalar-mass insertion. With the assumed power-counting (2.22), this contribution is formally of higher order. Finally, there is the fermion loop

$$D_5^{ij} = \frac{1}{2}\left(Y^{iIJ}Y_{jIJ} + Y_{iIJ}Y^{jIJ}\right)\left[-\frac{T^2}{12} - p^2 \frac{1}{(4\pi)^2 \epsilon_f}\right] . \tag{2.35}$$

The second coupling-constant term above is the hermitian conjugate. Also, above we use ϵ_b and ϵ_f to denote ϵ poles, albeit with some additional factors. They are defined in [30,34], and are

$$\frac{1}{\epsilon_b} = \frac{1}{\epsilon} + \text{Lb}, \quad \frac{1}{\epsilon_f} = \frac{1}{\epsilon} + \text{Lf}, \tag{2.36}$$

where Lb and Lf depend on the 4d-renormalization scale ($\bar{\mu}$) as well as the temperature (T) [30,34]. They are

$$\text{Lb} = \ln\left(\frac{\bar{\mu}^2}{T^2}\right) + 2\gamma_E - 2\ln(4\pi) , \quad \text{Lf} = \text{Lb} + 4\ln(2) , \tag{2.37}$$

where γ_E is the Euler-Mascheroni constant.

To perform the matching, we demand

$$\int d^4 x \langle R_i R_j \rangle_{4d} = T^{-1} \int d^3 x \langle R_i R_j \rangle_{4d} = \int d^3 x \langle R_i R_j \rangle_{3d} . \tag{2.38}$$

At leading order, 3d fields are rescaled by a factor $T^{-1/2}$. Next, consider the self-energies. One finds

$$\left(-\delta Z_{ij} p^2 - \delta\mu_{ij} - \mu_{ij} + \Pi(0)_{ij} + p^2 \Pi'(0)_{ij}\right)_{4d} = \left(-\delta Z_{3,ij} p^2 - \delta\mu_{3,ij} - \mu_{3,ij}\right)_{3d} , \tag{2.39}$$

where $\Pi_{ij}(p) = (D_1 + \cdots + D_5)_{ij}$. To avoid confusion we added a subscript 3 to three-dimensional quantities.

Here $-\delta Z_{ij} p^2 - \delta\mu_{ij}$ are 4d counterterms, and cancel all ϵ poles. The matching gives

$$\mu_{3,ij} = \mu_{ij} - \Pi(0)_{ij} , \quad \delta Z_{3,ij} = \delta Z_{ij} - \Pi'(0)_{ij} . \tag{2.40}$$

In this step all the Lorentz structure is stripped away, leaving mere group-theory factors.
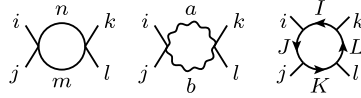
Fig. 2. Diagrams contributing to the thermal corrections to the scalar quartics. Indices correspond to the line notation in Fig. 1.

One-loop scalar quartics

Next, we consider scalar quartics. Before calculating the actual diagrams, it is useful to look at what happens on the 3d side. The relevant terms in the Lagrangian are

$$R_i R_j R_k R_l\, h_{ijkl}\,.\tag{2.41}$$

There are two contributions to the matching. First, the coupling constant, h, receives contributions from different loop orders

$$h_{ijkl} = h^0_{ijkl} + h^1_{ijkl} + \dots\tag{2.42}$$

Second, as shown in the previous section, 3d fields are renormalized. In terms of the scalar counterterm, $\delta Z_{3,ij}$, this renormalization is (to leading order) $R_{i(b)} = R_i + \frac{1}{2}\delta Z_{3,ij} R_j$. Combined, these considerations imply that the contribution from the 3d side is

$$\langle R_i R_j R_k R_l \rangle_{3d} = -h^0_{ijkl} - h^1_{ijkl}$$
$$-\frac{1}{2}\left(\delta Z_{3,im}h^0_{mjkl} + \delta Z_{3,jm}h^0_{imkl} + \delta Z_{3,km}h^0_{ijml} + \delta Z_{3,lm}h^0_{ijkm}\right)\,.\tag{2.43}$$

In the 4d theory, the first and second diagrams give

$$D^{ijkl}_1 = \frac{1}{2}\left(\lambda_{ijnm}\lambda_{nmkl} + \lambda_{iknm}\lambda_{nmjl} + \lambda_{ilnm}\lambda_{nmkj}\right)\frac{1}{16\pi^2\epsilon_b}\,,\tag{2.44}$$

$$D^{ijkl}_2 = \frac{1}{2}\left(H^{ab}_{ij}H^{ab}_{kl} + H^{ab}_{ik}H^{ab}_{jl} + H^{ab}_{il}H^{ab}_{kj}\right)\left[\frac{3}{(4\pi)^2\epsilon_b} - \frac{1}{8\pi}\right]\,.\tag{2.45}$$

Finally, the fermion contribution is

$$D^{ijkl}_3 = \frac{1}{2}\left(Y_{iIJ}Y^{jJK}Y_{kKL}Y^{lLI} + (jkl)\right)\frac{1}{(4\pi)^2\epsilon_f} + \text{h.c.}\tag{2.46}$$

where the terms (jkl) contain all permutations over indices j,k,l.

The matching is performed by demanding

$$\int \mathrm{d}^4 x \langle R_i R_j R_k R_l\rangle_{4d} = T^{-1}\int \mathrm{d}^3 x\langle R_i R_j R_k R_l\rangle_{4d} = \int \mathrm{d}^3 x\langle R_i R_j R_k R_l\rangle_{3d}\,.\tag{2.47}$$

In the light of the discussion at the beginning of this section, this implies

$$T\left(-\lambda_{ijkl} + \Lambda_{ijkl}\right) = -h^0_{ijkl} - h^1_{ijkl}$$
$$-\frac{1}{2}\left(\delta Z_{3,im}h^0_{mjkl} + \delta Z_{3,jm}h^0_{imkl} + \delta Z_{3,km}h^0_{ijml} + \delta Z_{3,lm}h^0_{ijkm}\right)\,.\tag{2.48}$$

The renormalized sum of the diagrams above was denoted as $\Lambda_{ijkl} = (D_1 + D_2 + D_3)_{ijkl}$ and one finds

$$h^0_{ijkl} = T\lambda_{ijkl},\tag{2.49}$$

$$h^1_{ijkl}T^{-1} = -\frac{1}{2}\left(\delta Z_{3,im}\lambda_{mjkl} + \delta Z_{3,jm}\lambda_{imkl} + \delta Z_{3,km}\lambda_{ijml} + \delta Z_{3,lm}\lambda_{ijkm}\right) - \Lambda_{ijkl}\,,\tag{2.50}$$

where $\delta Z_{3,km}$ was given in the previous section.

Higher loop levels

When performing dimensional reduction at NLO, two-loop contributions are required for scalar thermal masses. Here, we do not show details of the two-loop computation, as they are analogous to the one-loop computation presented above. Naturally the number of required diagrams is larger. Due to the use of integration-by-parts identities (IBP) [72–74] it can be shown that to the given order, the two-loop master sum-integrals factorize into the one-loop master integrals given in eqs. (2.28) and (2.29). While this streamlines the computation at NLO significantly, contributions at higher orders are non-factorizable. Nonetheless, DRalgo also implements a generic computation of two-loop thermal masses and other 3d parameters at NLO.

2.4. Beta functions and anomalous dimensions

DRalgo calculates beta functions and anomalous dimensions by renormalizing 4d parameters (cf. [64–66]). As an example, consider the scalar sector. We have renormalized our fields and couplings as

$$R_{i(b)} = Z_{ij}^{1/2} R_j = \left(\delta_{ij} + \frac{1}{2}\delta Z_{ij}\right) R_j \,, \tag{2.51}$$

$$R_{i(b)} R_{j(b)} R_{k(b)} R_{l(b)} \lambda_{ijkl(b)} = R_i R_j R_k R_l \left(\lambda_{ijkl} + \delta\lambda_{ijkl}\right) \bar{\mu}^{2\epsilon} \,. \tag{2.52}$$

On the 4d side, we assume the $\overline{\text{MS}}$ scheme, so δZ_{ij}, $\delta\mu_{ij}$, and $\delta\lambda_{ijkl}$ are chosen to *only* cancel ϵ poles.

We demand that bare parameters are independent of the RG-scale $\bar{\mu}$. To leading order in ϵ, we find ($t = \log\bar{\mu}$)

$$\partial_t \lambda_{ijkl} = -2\epsilon\lambda_{ijkl} \,, \quad \partial_t \delta Z_{ij} = -2\epsilon\delta Z_{ij}. \tag{2.53}$$

Using this we can find the anomalous dimensions by demanding that $\partial_t R_{i(b)} = 0$:

$$\implies \partial_t R_i = \epsilon Z_{ij} R_j \equiv \gamma_{ij} R_j \,, \tag{2.54}$$

where we have introduced the anomalous dimension γ_{ij}. Then, the quartic beta function is determined by

$$0 = \bar{\mu}^{-2\epsilon} \partial_t \lambda_{ijkl(b)} = 2\epsilon\left(\lambda_{ijkl} + \delta\lambda_{ijkl}\right) + \partial_t \delta\lambda_{ijkl} + \partial_t \lambda_{ijkl} \,,$$

$$\implies \partial_t \lambda_{ijkl} = -2\epsilon\lambda_{ijkl} + 2\epsilon\delta\lambda_{ijkl} \,. \tag{2.55}$$

Where we have used $\partial_t \delta\lambda_{ijkl} = -4\epsilon\delta\lambda_{ijkl}$; which follows from counting powers of couplings in $\delta\lambda_{ijkl}$. We identify $2\epsilon\delta\lambda_{ijkl}$ with the beta function for λ_{ijkl}.

This beta function can be written in an equivalent form by writing the counterterm, for λ, as the sum of a bare term and field renormalization:

$$\delta\lambda_{ijkl} = \bar{\lambda}_{ijkl} + \frac{1}{2}\left(\delta Z_{im}\lambda_{mjkl} + \delta Z_{jm}\lambda_{imkl} + \delta Z_{km}\lambda_{ijml} + \delta Z_{lm}\lambda_{ijkm}\right) \,. \tag{2.56}$$

Using this one finds

$$\partial_t \lambda_{ijkl} = -2\epsilon\lambda_{ijkl} + 2\epsilon\bar{\lambda}_{ijkl} - (\gamma\lambda)_{(ijkl)} \,, \tag{2.57}$$

where the last term denotes all possible contractions between one index of λ_{ijkl} and one index of γ_{ij}. Other beta functions are calculated analogously.

All gauge beta functions in DRalgo are defined with couplings squared, *viz.* $\partial_t g^2$. Conversely beta functions for masses, scalar couplings, and Yukawa couplings are defined as linear in the parameters, i.e. $\partial_t \lambda$ and so forth. These conventions are also shown in the DRalgo output.

In the 3-dimensional theory only scalar masses have non-zero beta functions.

3. Installation and running

This section explains how to install DRalgo and presents a tutorial based on the Abelian-Higgs model.

3.1. Installation

The current version of DRalgo, 1.0, is installed by placing all the source files either in the applications folder $UserBaseDirectory/Applications or running Mathematica from the package root directory ./DRalgo.

The required source files are outlined in Fig. 3. To create model files, DRalgo uses functions from GroupMath [75]; see [75] for its installation. This tutorial uses GroupMath Version 1.1.2 but GroupMath is not required for the dimensional reduction itself. If a model file is already available, it can be loaded independently of GroupMath (cf. sec. Q.1). Since GroupMath is an external package, any use of the model-creation features in DRalgo should be accompanied by a citation of [75].

To load DRalgo from the Mathematica folder $UserBaseDirectory/Applications, the following commands need to be executed:

```
SetDirectory[NotebookDirectory[]];
$LoadGroupMath=True;
<<DRalgo`
```

Subsequently, we demonstrate the definition of models, the dimensional reduction, and the calculation of the two-loop effective potential. All steps are mirrored in the accompanying ./examples/ah.m.

```
DRalgo
  ├─ DRalgo.m ........................................................................package file
  ├─ Debye.m ......................................................Creates and organizes temporal scalars
  ├─ EffPot.m .............................................Computation of the effective potential in the EFT
  ├─ HardToSoft.m ...............................................................Matching hard to soft
  ├─ SoftToUS.m ...........................................................Matching soft to ultrasoft
  ├─ ModelCreation.m ...............................................................Creating a model
  ├─ examples/ ...............................................................Model example library
  │    ├─ ah.m..........................................................................Abelian-Higgs model
  │    ├─ sm.m..................................................................................Standard Model
  │    ├─ xsm.m.................................................................Standard Model and one real scalars
  │    ├─ 2xsm.m...............................................................Standard Model and two real scalars
  │    ├─ cxsm.m..............................................................Standard Model and one complex scalar
  │    ├─ 2hdm.m................................................................... Two-Higgs Doublet model
  │    ├─ 3hdm.m.................................................................Three-Higgs Doublet model
  │    ├─ htm.m.................................................Standard Model and one real SU(2) triplet
  │    ├─ SU5.m..........................................................................SU(5) model
  │    ├─ LRSymmetric.m............................................Left-Right symmetric model
  │    ├─ WessZumino.m................................................Wess-Zumino model
  │    └─ smZp.m..............................................Standard Model and Z′-bosons
  ├─ README.md
  └─ LICENSE
```

Fig. 3. Outline of the structure of DRalgo. The most relevant files for this article are highlighted. Specific models are discussed in sec. 3 for ah.m and sec. 4 for 2hdm.m.

3.2. Model implementation

As an example, we consider the Abelian-Higgs model with the Euclidean Lagrangian

$$\mathcal{L} = \frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \left(D^\mu \phi\right)^\dagger \left(D_\mu \phi\right) + V\left(\phi, \phi^\dagger\right), \tag{3.1}$$

$$V\left(\phi, \phi^\dagger\right) = m^2 \phi^\dagger \phi + \lambda \left(\phi^\dagger \phi\right)^2, \tag{3.2}$$

where the covariant derivative is $D_\mu = \partial_\mu - ig_1 Y_\phi A_\mu$ and the corresponding field-strength tensor is $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$. Here, the field A_μ is an U(1) gauge field with gauge coupling g_1 and ϕ is a complex scalar field charged under U(1) with general charge Y_ϕ.

The zero, or temporal, component of the vector field receives a thermally induced mass in the 3d theory. This temporal scalar transforms as a singlet under U(1) and has a Lagrangian

$$\mathcal{L}_{\text{temporal}} = \frac{1}{2}(\partial_i A_0)^2 + \frac{1}{2}\mu\text{sqU1}\,A_0^2 + \frac{1}{4!}\lambda\text{VLL[1]}\,A_0^4 + \frac{1}{2}\lambda\text{VL[1]}\,A_0^2 \phi^\dagger \phi, \tag{3.3}$$

where indices $i \in \{1, \dots, d\}$ are spatial. Here, μsqU1 is the temporal-scalar Debye mass (squared), $\lambda\text{VLL[1]}$ is the self-interaction coupling, and $\lambda\text{VL[1]}$ is the portal coupling to ϕ. The notation above is aligned with the systematic notation of DRalgo. For earlier studies of this model in the literature, we refer to [36–38,76,77].

To create a model in DRalgo, we have to specify the gauge group and the scalar representation. In DRalgo this is done via

```
Group={"U1"};
CouplingName={g1};
RepAdjoint={0};
Higgs={{Yϕ},"C"};
RepScalar={Higgs};
RepFermion={};
```

Here, the adjoint (vector) representation is trivial for the Abelian Higgs model. In the definition of the Higgs scalar $\{\{Y\phi\}, "C<R>"\}$, Y_ϕ denotes the gauge charge. One additional argument is passed depending on if the scalar has a real (R) or a complex (C) representation. In this simple example, fermions are absent and the corresponding bracket is empty. While not relevant for this model, note that fermions are never part of the 3d EFT Lagrangian as they are integrated out during the first step of the dimensional reduction. The model-input from the gauge sector is generated with the command

```
{gvvv,gvff,gvss,λ1,λ3,λ4,μij,μIJ,μIJC,Ysff,YsffC}=
   AllocateTensors[Group,RepAdjoint,CouplingName,RepFermion,RepScalar];
```

The tensors listed above have the following correspondence

`gvvv`	structure constants
`gvff`	vector-fermion trilinear couplings
`gvss`	vector-scalar trilinear couplings
`λ1`	scalar tadpole couplings
`λ3`	cubic couplings
`λ4`	quartic couplings
`μij`	scalar-mass matrix
`μIJ`, `μIJC`	fermion-mass matrices
`Ysff`, `YsffC`	Yukawa couplings

and their purpose is described in depth in sec. 2. By default all these tensors, except for the gauge ones, are empty. While their order on the left hand side of `AllocateTensors` is fixed, the above naming of $\{$`gvvv`,...,`YsffC`$\}$ is arbitrary.

The next task is to specify the scalar potential. Let us start with the mass matrix. There is only one allowed term: $\phi\phi^\dagger$. This term is selected via

```
InputInv={{1,1},{True,False}};
MassTerm1=CreateInvariant[Group,RepScalar,InputInv][[1]]//Simplify;
VMass=msq*MassTerm1; (* corresponds to a term ½m²φφ† *)
μij=GradMass[VMass];
```

To understand the syntax, recall that we defined our scalar as `RepScalar={Higgs}`. Then $\{$`1,1`$\}$ specifies that we are after a term with two (`RepScalar[[1]]`) ϕ fields. For general models, the index `1` can be replaced by any index in `RepScalar` as defined by the user. In addition, $\{$`True,False`$\}$ specifies that we are after the $\phi\phi^\dagger$ term and conversely $\{$`True,True`$\}$ corresponds to a $\phi\phi$ term. The latter term, however, is not allowed in this example due to gauge invariance. Since this example exhibits merely one U(1) invariant combination of ϕ and $\phi^\dagger$, we specify `[[1]]` in the `CreateInvariant` output.

For the quartic tensor only one possible term arises, namely $(\phi\phi^\dagger)^2$. We already created a $(\phi\phi^\dagger)$ term above, which can be reused to find the quartic:

```
VQuartic=λ*MassTerm1^2;
λ4=GradQuartic[VQuartic]
```

To see how quartic and mass terms are composed of scalar-field components, one can inspect the variables `VMass` and `VQuartic`.

This completes the model implementation, and we now have all ingredients to do the dimensional-reduction step.

3.3. Running the dimensional-reduction algorithm

First we need to load the model with the command

```
ImportModelDRalgo[Group,gvvv,gvff,gvss,λ1,λ3,λ4,μij,μIJ,μIJC,Ysff,YsffC,Verbose->False];
```

Since the gauge group was previously defined, the corresponding Debye masses are automatically generated and named. The option `Verbose->False` disables all progress messages.

The actual dimensional reduction is performed with the command

```
PerformDRhard[];
```

This calculates all thermal masses, effective couplings, β-functions and anomalous dimensions. For example, gauge and scalar couplings are given by

```
PrintCouplings[]
```
$$\left\{\text{g13d}^2 \to g_1^2 T - \frac{g_1^4 \text{Lb} T Y_\phi^2}{48\pi^2}, \; \lambda 3\text{d} \to \frac{T(g_1^4 Y_\phi^4 (2-3\text{Lb}) + 6g_1^2 Y_\phi^2 \lambda \text{Lb} + 2\lambda(8\pi^2 - 5\lambda \text{Lb}))}{16\pi^2}\right\}$$

where the output is a replacement rule. The variables `Lb` and `Lf` are given in eq. (2.37), and depend on the 4d-renormalization scale ($\bar\mu$) as well as the temperature (T) [30,34]. Further information about the functional basis of the effective parameters is collected in sec. Q.11. The corresponding definitions of constants, along with other shorthand notations, are shown with the command

```
PrintConstants[]
```

The above RG-scale $\bar\mu$ is the hard matching scale, and should be chosen as $\bar\mu \sim T$ to avoid large logarithms.[6] Effective mass parameters are divided into scalar and temporal-scalars. The scalar masses are given by

```
PrintScalarMass["LO"]
PrintScalarMass["NLO"]
```

---

[6] In `DRalgo` the renormalization scales are denoted by $\mu = \bar\mu$ for the hard scale and $\mu 3 = \bar\mu_3$ for the soft scale.

Here, the LO command prints tree-level masses together with one-loop thermal contributions; the NLO command prints two-loop thermal masses. By default these NLO masses contain the running from the matching scale ($\bar{\mu}$) to an arbitrary 3d scale ($\bar{\mu}_3$).[7] The resulting Debye masses are displayed by

```
PrintDebyeMass["LO"]
PrintDebyeMass["NLO"]
```

Mixed temporal-scalar couplings (to NLO) are printed with

```
PrintTemporalScalarCouplings[]
```

Such temporal-scalar couplings are denoted as λVLL[a] for V^4 couplings, λVVSL[a] for V^2S couplings and λVL[a] for V^2S^2 couplings, where S represents scalar fields and V temporal scalar fields. The index 'a' systematically labels all linearly independent terms.

The coefficient of the unit operator [35], or the hard-mode contribution to the symmetric-phase pressure, is given by

```
PrintPressure["LO"]
PrintPressure["NLO"]
PrintPressure["NNLO"]
```

Here, LO corresponds to one-loop, NLO to two-loop, and NNLO to three-loop level. The LO result describes the pressure of an ideal gas, and is given by $P_{\text{LO}}(T) = N\frac{\pi^2}{90}T^4$, where N represents the number of degrees of freedom. For the Abelian Higgs model $N = 4$ with 2 from the photon and 2 from the complex scalar.

DRalgo also provides anomalous dimensions and beta functions in the parent 4d theory; see sec. 2.4. Beta functions are printed with the command

```
BetaFunctions4D[]
```

Next, to find anomalous dimensions we first need to specify for which particles we want them. As DRalgo stores everything in tensor form, we need to find the positions of all particles

```
PosScalar=PrintScalarRepPositions[];
```

To determine the anomalous dimension of e.g. ScalarRep[[1]], we would write

```
AnomDim4D["S",{PosScalar[[1]],PosScalar[[1]]}]
```

See sec. Q.4 for how to obtain anomalous dimensions for vector-bosons and fermions.

3.4. Integrating out temporal scalars

Temporal scalars are often heavy compared to the fields driving the phase transition, and can be integrated out [30,34]. It should be stressed that this is a user-dependent optional step. The resulting EFT is said to describe ultrasoft physics. The command is

```
PerformDRsoft[{}];
```

Couplings at the ultrasoft scale are given by

```
PrintCouplingsUS[];
```

And effective scalar masses are given by

```
PrintScalarMassUS["LO"];
PrintScalarMassUS["NLO"];
```

Finally, β-functions for the ultrasoft masses are provided with the command

```
BetaFunctions3DUS[];
```

In the 3-dimensional theory only scalar masses have non-zero beta functions.

This concludes our tutorial of the dimensional reduction of the Abelian-Higgs model.

---

[7] There are different ways to express the NLO masses. We refer to sec. Q.11 for further details.

3.5. Two-loop effective potential

Once the model is loaded, we can calculate the effective potential. This can be done by the user, either in the soft or the ultrasoft theory. Alternatively, DRalgo can calculate the effective potential in the ultrasoft theory. First, we need to create field-dependent masses which requires to specify a vacuum expectation value direction:

```
DefineNewTensorsUS[μij,λ4,λ3,gvss,gvvv];
φvev={0,φ}//SparseArray;
DefineVEVS[φvev];
```

Here the vacuum expectation value is in the second, imaginary, Higgs component.

For the calculation to proceed, all mass matrices must be diagonal. If not, an error message is printed and it is up for the user to diagonalize the matrices; see section Q.16 for an example. For the model at hand, the mass matrix is diagonal, which the user can confirm by printing the field-dependent masses

```
FieldMasses=PrintTensorsVEV[]
```

The effective potential is calculated via

```
CalculatePotentialUS[]
```

The results are given by

```
PrintEffectivePotential["LO"]
PrintEffectivePotential["NLO"]
PrintEffectivePotential["NNLO"]
```

Here, LO refers to the tree-level, NLO to the one-loop, and NNLO to the two-loop effective potential. Note that all results are given in Landau gauge. In the (two-loop) NNLO part, the renormalization scale is denoted by $\mu 3 = \bar{\mu}_3$.

This completes the first tutorial on quick installation and running. Output from the 3d EFT matching relations can be implemented either to non-perturbative lattice codes, or perturbative analyses in terms of the effective potential. While such implementations are left to the user, appendix A displays how to interface DRalgo output in a Mathematica implementation that determines selected thermodynamic quantities for the Abelian-Higgs model.

4. Implementing beyond the Standard Model theories

In this section, we demonstrate the use of DRalgo for BSM theories based on the example of the Two-Higgs doublet model (2HDM).

4.1. Two-Higgs doublet model with fermions

Consider the 2HDM potential [78,79]

$$
\begin{aligned}
V(\phi_1, \phi_2) = {} & m_1^2 \phi_1 \phi_1^\dagger + m_2^2 \phi_2 \phi_2^\dagger - (m_{12}^2 \phi_1 \phi_2^\dagger + \text{h.c.}) + \lambda_1 (\phi_1 \phi_1^\dagger)^2 + \lambda_2 (\phi_2 \phi_2^\dagger)^2 \\
& + \lambda_3 (\phi_1 \phi_1^\dagger)(\phi_2 \phi_2^\dagger) + \lambda_4 (\phi_1 \phi_2^\dagger)(\phi_2 \phi_1^\dagger) + \frac{\lambda_5}{2} \left[(\phi_1 \phi_2^\dagger)^2 + (\phi_2 \phi_1^\dagger)^2 \right] \\
& + \left\{ \left[\lambda_6 (\phi_1 \phi_1^\dagger) + \lambda_7 (\phi_2 \phi_2^\dagger) \right] (\phi_1^\dagger \phi_2) + \text{h.c.} \right\} .
\end{aligned}
\tag{4.1}
$$

The scalars are SU(2) doublets with hypercharge $Y_\phi = 1$. This means that the covariant derivative is given by[8]

$$
\mathcal{D}_\mu \phi_{1,2} = \left(\partial_\mu - i g_1 \frac{Y_\phi}{2} B_\mu - i g_2 \frac{\tau^a}{2} A_\mu^a \right) \phi_{1,2} ,
\tag{4.2}
$$

where τ^a are Pauli matrices. We further assume that all Standard-Model fermions are present, and consider Yukawa interactions of the form

$$
\mathcal{L}_{\text{Yuk}} = y_i \overline{q}_L \phi_i^\dagger q_R + \text{h.c.}
\tag{4.3}
$$

We define the group structure and scalar representations via

```
Group={"SU3","SU2","U1"};
RepAdjoint={{1,1},{2},0};
HiggsDoublet1={{{0,0},{1},1/2},"C"};
HiggsDoublet2={{{0,0},{1},1/2},"C"};
RepScalar={HiggsDoublet1,HiggsDoublet2};
CouplingName={g3,g2,g1};
```

---

[8] We use the convention that $(\phi_1 \phi_1^\dagger) = (\phi_1^\dagger \phi_1) = \phi_1^I \phi_1^{\dagger,I}$, where $I = 1, 2$.

All representations are specified according to their Dynkin coefficients (see e.g. [75,80,81]). For example, gauge bosons transform as $(1, 1)$, or an octet, under SU(3). Next we need to create the fermions. Let us start by creating a single generation

```
Rep1={{{1,0},{1}, 1/6},"L"};  (*q_L*)
Rep2={{{1,0},{0}, 2/3},"R"};  (*u_R*)
Rep3={{{1,0},{0},-1/3},"R"};  (*d_R*)
Rep4={{{0,0},{1},-1/2},"L"};  (*ℓ_L*)
Rep5={{{0,0},{0},-1},"R"};  (*e_R*)
RepFermion1Gen={Rep1,Rep2,Rep3,Rep4,Rep5};
```

In the definition of a fermion representation $\{\{1,0\},\{1\},Y/2\},$ "L<R>"$\}$ the last argument depends on if the fermion left-handed (`L`) or right-handed (`R`) and `Y` is the corresponding hypercharge. For the Standard Model, we have

$$Y_q = \frac{1}{3}, \quad Y_u = \frac{4}{3}, \quad Y_d = -\frac{2}{3}, \quad Y_\ell = -1, \quad Y_e = -2. \tag{4.4}$$

The above notation identifies `Rep1` as the left-handed quark doublet, `Rep2` as the right-handed up-quark and so forth. The extra factor of $\frac{1}{2}$ for hypercharges in the definition of e.g. `Rep1` arises from the definition of the covariant derivative similar to eq. (4.2).

Additional fermion-families can be added by grouping multiple instances of `RepFermion1Gen` together:

```
RepFermion3Gen={RepFermion1Gen,RepFermion1Gen,RepFermion1Gen}//Flatten[#,1]&;
```

where the number of generations is chosen to be $n_f = 3$. When creating Yukawa interactions the user decides which index contains which family. For example, above we stacked all generations after each other in `RepFermion3Gen`. Hence, in `RepFermion3Gen` indices 1–5 correspond to the first generation, indices 6–10 to the second, and indices 11–15 to the last. Thus the user could use index 1, 6, or 11 as the top quark. An alternative way to add an arbitrary number of fermion families n_f is given in sec. Q.9.

Next, to create the tensors, we write

```
{gvvv,gvff,gvss,λ1,λ3,λ4,μij,μIJ,μIJC,Ysff,YsffC}=
    AllocateTensors[Group,RepAdjoint,CouplingName,RepFermion3Gen,RepScalar];
```

The mass terms in the potential in eq. (4.1) are

$$V(\phi_1, \phi_2) \supset m_1^2 \phi_1 \phi_1^\dagger + m_2^2 \phi_2 \phi_2^\dagger - (m_{12}^2 \phi_1 \phi_2^\dagger + \text{h.c.}), \tag{4.5}$$

and due to the presence of two Higgs doublets, we need to specify the doublet for each term:

```
InputInv={{1,1},{True,False}};  (*φ_1φ_1†*)
MassTerm1=CreateInvariant[Group,RepScalar,InputInv][[1]]//Simplify;
InputInv={{2,2},{True,False}};  (*φ_2φ_2†*)
MassTerm2=CreateInvariant[Group,RepScalar,InputInv][[1]]//Simplify;
InputInv={{1,2},{True,False}};  (*φ_1φ_2†*)
MassTerm3=CreateInvariant[Group,RepScalar,InputInv][[1]]//Simplify;
InputInv={{2,1},{True,False}};  (*φ_2φ_1†*)
MassTerm4=CreateInvariant[Group,RepScalar,InputInv][[1]]//Simplify;
```

The mass matrix is then generated by the `GradMass[]` command:

```
VMass=(
    +m1*MassTerm1
    +m2*MassTerm2
    -(m12R+I*m12I)(MassTerm3)
    -(m12R-I*m12I)(MassTerm4)
    );
μij=GradMass[VMass]//Simplify;
```

We allowed m_{12}^2 to be complex, with real part `m12R`, and imaginary part `m12I`. For the scalar quartics in eq. (4.1) the corresponding part of the potential is

$$V(\phi_1, \phi_2) = \lambda_1 (\phi_1 \phi_1^\dagger)^2 + \lambda_2 (\phi_2 \phi_2^\dagger)^2 + \lambda_3 (\phi_1 \phi_1^\dagger)(\phi_2 \phi_2^\dagger)$$
$$+ \lambda_4 (\phi_1 \phi_2^\dagger)(\phi_2 \phi_1^\dagger) + \frac{\lambda_5}{2} \left[(\phi_1 \phi_2^\dagger)^2 + (\phi_2 \phi_1^\dagger)^2 \right]$$
$$+ \left\{ \left[\lambda_6 (\phi_1 \phi_1^\dagger) + \lambda_7 (\phi_2 \phi_2^\dagger) \right] (\phi_1^\dagger \phi_2) + \text{h.c.} \right\}, \tag{4.6}$$

for which we already created all the building blocks. Thus, the quartics can be constructed as

```
QuarticTerm1=MassTerm1^2;  (*(φ_1φ_1†)^2*)
QuarticTerm2=MassTerm2^2;  (*(φ_2φ_2†)^2*)
QuarticTerm3=MassTerm1*MassTerm2;  (*(φ_1φ_1†)(φ_2φ_2†)*)
```

```
QuarticTerm4=MassTerm3*MassTerm4;  (*(ϕ₁ϕ₂†)(ϕ₂ϕ₁†)*)
QuarticTerm5=(MassTerm3^2+MassTerm4^2);  (*(ϕ₁ϕ₂†)²+(ϕ₂ϕ₁†)²*)
QuarticTerm6=MassTerm1*MassTerm3+MassTerm1*MassTerm4;  (*(ϕ₁ϕ₁†)[(ϕ₁ϕ₂†)+(ϕ₂ϕ₁†)]*)
QuarticTerm7=MassTerm2*MassTerm3+MassTerm2*MassTerm4;  (*(ϕ₂ϕ₂†)[(ϕ₁ϕ₂†)+(ϕ₂ϕ₁†)]*)
```

Consequently, the quartic tensor itself is defined as

```
VQuartic=(
   +λ1H*QuarticTerm1
   +λ2H*QuarticTerm2
   +λ3H*QuarticTerm3
   +λ4H*QuarticTerm4
   +λ5H/2*QuarticTerm5
   +λ6H*QuarticTerm6
   +λ7H*QuarticTerm7
   );
λ4=GradQuartic[VQuartic];
```

For simplicity, we assumed above that λ5H, λ6H, and λ7H are real. One can allow for complex couplings by adding them, and their conjugates, directly in `QuarticTerm6` and `QuarticTerm7`. To create a complex coupling λ6H, we would write

```
QuarticTerm6=(λ6HR+I*λ6HI)*MassTerm1*MassTerm3+(λ6HR-I*λ6HI)*MassTerm1*MassTerm4;
```

For Yukawa couplings, we only consider the top-quark coupling. If we choose the first family to contain the top-quark, then a Yukawa term $\sim \phi^\dagger \overline{q}_L u_R$ would involve fermion representations number 1 and 2. When defining a Yukawa interaction, representations should be specified in the order: scalar, first fermion, second fermion. With this in mind, the Yukawa coupling of the first Higgs doublet is[9]

```
InputInv={{1,1,2},{False,False,True}};  (*ϕ₁†q̄_L u_R*)
YukawaDoublet1=CreateInvariantYukawa[Group,RepScalar,RepFermion3Gen,InputInv][[1]]//Simplify;
```

Here, `{1,1,2}` specifies

First Higgs doublet $\times$ Left-handed top-quark $\times$ Right-handed top-quark,

and `{False,False,True}` specifies that the Higgs and left-handed quarks are conjugated. The coupling to the second Higgs doublet is

```
InputInv={{2,1,2},{False,False,True}};  (*ϕ₂†q̄_L u_R*)
YukawaDoublet2=CreateInvariantYukawa[Group,RepScalar,RepFermion3Gen,InputInv][[1]]//Simplify;
```

The complete Yukawa tensor is

```
Ysff=-GradYukawa[yt1*YukawaDoublet1+yt2*YukawaDoublet2];
```

For the sake of generality, we assumed that both Higgs doublets couple to the top-quark. Note that this is not allowed for a realistic model due to flavor-changing-neutral-current constraints.

Finally, assuming that the Yukawa couplings are real, we can define

```
YsffC=Simplify[Conjugate[Ysff//Normal],Assumptions->{yt1>0,yt2>0}]//SparseArray;
```

Above we did not consider Yukawa couplings between different generations but such couplings can be added. To wit, when we defined `RepFermion3Gen` we put the first generation on indices 1–5; the second on indices 6–10; and the third on indices 11–15. To define a coupling between, e.g. the left-handed top quark (assumed to reside in generation 1) and the right-handed charm quark (assumed to reside in generation 2), we would write

```
InputInv={{1,1,7},{False,False,True}};
YukawaTopCharm=CreateInvariantYukawa[Group,RepScalar,RepFermion3Gen,InputInv]//Simplify;
```

With the model implementation complete, all dimensional-reduction commands are identical to those of the Abelian-Higgs model. We refer to [82–84] for earlier results in the literature.

Integrating out temporal scalars

For the Abelian-Higgs case in sec. 3 we only had one scalar field, whereas with the 2HDM we have two. When integrating out temporal scalars (cf. sec. 3.4), we have two options. First, all scalars are light and we have two active, dynamical doublets. Second, one doublet is heavy and is integrated out when going from the soft to the ultrasoft scale. The following commands can be used after the reduction from the hard to the soft scale have been performed i.e. `PerformDRhard[];`

---

[9] Here we assume that the top quark resides in the first generation. This has nothing to do with how the top-quark is usually placed in the third generation. Rather, we here place it in the first for simplicity as normally only the top-quark has a sizeable Yukawa coupling.

For the first case, the corresponding command is

```
PerformDRsoft[{}];
```

For the second case, we assume that the second doublet is heavy. It can then be integrated out via

```
PerformDRsoft[{5,6,7,8}];
```

To understand the above syntax, note that the indices of all representations are given by

```
PosScalars=PrintScalarRepPositions[];
```

So the second Higgs doublet resides at `PosScalars[[2]]=5;;8`.

There is one complication when integrating out one of the Higgs doublets. Namely, generally the two doublets mix through the m_{12}^2 term. For a complete treatment the mass matrix needs to be diagonalized before the heavy doublet can be integrated out. This lies beyond DRalgo and is an optional step for the user. Instead, the code by default assumes that m_{12}^2 is small – of $\mathcal{O}(g^2 T)$ in power counting – and performs the diagonalization perturbatively to first order in m_{12}^2.

5. Miscellaneous features

In this section, we discuss specific features of the software on a case-by-case basis.

5.1. User-options and features

Lower-order dimensional reduction for speed

The user has several options to control what is calculated and how the code operates. For example, if the model has many degrees of freedom, the user might wish to save running-time and only calculate one-loop thermal masses and couplings. This works by specifying `Mode->1` when loading the model

```
Group={"U1"};
ImportModelDRalgo[Group,gvvv,gvff,gvss,λ1,λ3,λ4,μij,μIJ,μIJC,Ysff,YsffC,Mode->1];
```

The default is `Mode->2`, in which case everything is calculated to NLO. And for the most bare-bone/fast option select `Mode->0`; in this case only one-loop thermal masses are calculated.

5.2. Model-treatment in the code

DRalgo works by factorizing all group and Lorentz algebra. The Lorentz algebra is hard-coded while the group algebra is supplied by the user. All particles are indexed by the order they appear. If the user has an SU(3) × SU(2) model with gauge bosons, the vectors in the SU(3) group have 8 components, while those of the SU(2) group have 3. The code indexes these components by $a = \left(8_{SU(3)}^{\alpha}, 3_{SU(2)}^{\beta}\right)$. Therein, α runs from 1–8, β from 1–3, and a from 1–11. For example, the Debye mass tensor μ_D^{ab} stores the information of both groups. For most cases this mass tensor is diagonal.

The scalars are treated differently since DRalgo only deals with real scalar components. For example, a complex scalar Φ is rewritten $\Phi = \frac{1}{\sqrt{2}}(\phi + i\psi)$. Hence the scalar indices would in this case be $i = (\phi, \psi)$. Further, the scalar-mass matrix is stored as $\mu_{S,ij}$. For example, a $m^2\phi^2$ mass term resides at $\mu_{S,11}$. Consider now a complex representation with n scalar components labelled by I, J. In a complex basis vector-scalar-scalar trilinear couplings are of the form $A_\mu \left(G_{IJ}^a \partial^\mu \Phi_I \Phi_J^* + G_{IJ}^{a,*} \partial^\mu \Phi_I^* \Phi_J\right)$, where G_{IJ}^a are representation matrices. To convert to a real basis each scalar component is split as $\Phi_I = \frac{1}{\sqrt{2}}(\phi_I + i\psi_I)$.[10] The components are then reordered with the real components first: $i = (\phi_I, \psi_I)$, so $i = 1, \ldots, 2n$. If we call the scalars in the real basis φ_i, our vector-scalar-scalar trilinear couplings become

$$A_\mu g_{ij}^a \partial^\mu \varphi_i \varphi_j, \qquad g_{ij}^a = \begin{pmatrix} \mathrm{Im}(G_{IJ}^a) & \mathrm{Re}(G_{IJ}^a) \\ -\mathrm{Re}(G_{IJ}^a) & \mathrm{Im}(G_{IJ}^a) \end{pmatrix}. \tag{5.1}$$

Thus, g_{ij}^a is automatically antisymmetric under $i \leftrightarrow j$.

As an example, consider a U(1) theory with a complex scalar. The vector-scalar-scalar coupling is $igA_\mu \left(\partial^\mu \Phi \Phi^* - \partial^\mu \Phi^* \Phi\right)$. In a real basis $\Phi = \frac{1}{\sqrt{2}}(\phi + i\psi)$, this becomes

$$gA_\mu \partial^\mu (\phi, \psi) \begin{pmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} \phi \\ \psi \end{pmatrix}, \quad \text{and} \quad g_{ij} = g \begin{pmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix}. \tag{5.2}$$

---

[10] Such a Cartesian basis is useful as all scalar-vector trilinear couplings are automatically anti-symmetric.

5.3. Frequently asked questions

Below, we discuss various questions and problems the user might have encountered.

Q.1. How to save and load my model?

Saving and loading a model is straightforward with `DRalgo` built-in functions. Once a model is created, and loaded with `Import-ModelDRalgo` (see the `.m` files in the example folder), it can be saved by writing

```
SaveModelDRalgo[ModelInfo,"<modelname>.txt"],
```

Here, `ModelInfo` is a string and should contain information about the authors, `DRalgo` version, and citations to relevant articles. See `./examples/ah.m` or `./examples/2hdm.m` for some examples. To load the model write

```
{Group,gvvv,gvff,gvss,λ1,λ3,λ4,μij,μIJ,μIJC,Ysff,YsffC}=LoadModelDRalgo["<modelname>.txt"];
```

The loaded model can be your own, or perhaps one of the provided `DRalgo` model repository. We also encourage you to make your own models available for the community which is possible by submitting the model file via the Issue Tracker on https://github.com/DR-algo/DRalgo. The model will then be verified and added to the model repository. When submitting a model, please refer to a paper or explicitly write out the Lagrangian in an accompanying notebook.

Q.2. How do I order semi-simple groups?

If the user-defined model contains multiple groups, the groups and corresponding representations must be ordered as: SO(n), SU(n), SP(n), G_2, F_4 E_6, E_7, E_8, U(1). For example, if the user has a model with gauge group $G = SU(5) \otimes SO(10) \otimes E_6 \otimes U(1)$, the model-input should be defined as

```
Group={"SO10","SU5","E6","U1"};
```

The internal order of groups is not restricted. Hence, for a group such as $G = SU(3) \otimes SU(2)$, the following definitions are equivalent

```
Group={"SU3","SU2"};
Group={"SU2","SU3"};
```

Q.3. How do I check that my model is anomaly free?

Once you have defined your model, the anomaly-free condition is that

```
Table[Tr[(a.b+b.a).c],{a,gvff},{b,gvff},{c,gvff}]
```

vanishes identically. Since this condition is not automatically fulfilled for general, non-numeric, U(1) charges, it is the responsibility of the user to choose U(1) charges such that all anomalies cancel.

Q.4. How do I calculate anomalous dimensions?

First find the position of all representations:

```
PosScalar=PrintScalarRepPositions[];
PosVector=PrintGaugeRepPositions[];
PosFermion=PrintFermionRepPositions[];
```

All anomalous dimensions are then found via

```
Table[AnomDim4D["S",{i,j}],{i,PosScalar},{j,PosScalar}]
Table[AnomDim4D["V",{i,j}],{i,PosVector},{j,PosVector}]
Table[AnomDim4D["F",{i,j}],{i,PosFermion},{j,PosFermion}]
```

For more details regarding anomalous dimensions and beta functions cf. sec. 2.4.

Q.5. Temporal scalar mixing

In most models the temporal-scalar masses are diagonal. However, in cases with multiple U(1) groups there can be mixing. In such a case, the code automatically recognizes this. For example, if the group is

```
Group={"U1","U1","U1"};
```

the code creates the mixed U(1) masses with the convention that μU1Mix1 is the mixing between groups 1 and 2, μU1Mix2 is the mixing between groups 1 and 3, and μU1Mix3 is the mixing between groups 2 and 3. These masses are displayed with the `PrintDe-byeMass` command. See the `./examples/SMZp.m` for an example.

Q.6. What if I miss some couplings?

By default DRalgo assumes that the user has defined all couplings allowed by symmetry in their 4d theory. The code runs even though some couplings are missed. For example, consider the 2HDM with scalar potential as in eq. (4.1) but without λ_6 and λ_7 couplings. For a $\phi_1 \leftrightarrow \phi_2$ symmetric model no λ_6/λ_7-type scalar quartic couplings are induced. However, if the model breaks this symmetry by e.g. a Yukawa sector, then λ_7 and λ_6 couplings are generated at one-loop. In this case DRalgo would still calculate these induced couplings; with the crux that only the $\lambda_1, \ldots, \lambda_5$ couplings are printed by the PrintCouplings[] command, while the induced λ_7 and λ_6 couplings must be found manually via the PrintTensorDRalgo[] command. In addition, DRalgo automatically alerts the user with a message if some new couplings are generated at one-loop. This is a cross-check that no couplings are forgotten.

Q.7. Can I run DRalgo without specifying the representation?

For general groups no. However, it is possible for the user to specify arbitrary (non-numeric) U(1) charges. With non-numeric charges the code does not check that various quartic or Yukawa terms are allowed. Hence, it is the responsibility of the user to ensure gauge invariance.

Q.8. How large representations can I use?

In principle any group and representations can be used. In practice the code slows down for huge representations. The code has been tested on general models with $\sim$100–200 components in a given representation. For example an SO(10) model with a 120-dimensional scalar. Since DRalgo was purposely written to deal with any quartic and Yukawa sector, these are often the bottlenecks. If the user wants to omit two-loop contributions, significantly larger representations are possible. Further still, if the user only wants one-loop thermal masses, almost any model (within reason) can be run in quick order.

Q.9. Can I include an arbitrary number of fermion generations?

Yes, take for example the 2HDM. As described in sec. 4.1, a single family of SM fermions is defined via

```
Rep1={{{1,0},{1}, 1/6},"L"}; (*q_L*)
Rep2={{{1,0},{0}, 2/3},"R"}; (*u_R*)
Rep3={{{1,0},{0},-1/3},"R"}; (*d_R*)
Rep4={{{0,0},{1},-1/2},"L"}; (*ℓ_L*)
Rep5={{{0,0},{0},-1},"R"}; (*e_R*)
RepFermion1Gen={Rep1,Rep2,Rep3,Rep4,Rep5};
```

Defining Yukawa and scalar couplings proceed as before, and the model is loaded with

```
ImportModelDRalgo[Group,gvvv,gvff,gvss,λ1,λ3,λ4,μij,μIJ,μIJC,Ysff,YsffC,Verbose->False];
```

The user can then define $n_{\rm f} = $ nF fermion families by writing[11]

```
PosFermion=PrintFermionRepPositions[];
FermionMat=Table[{nF,i},{i,PosFermion}];
DefineNF[FermionMat]
```

These commands should be used before running

```
PerformDRhard[];
```

Adding $n_{\rm f}$ fermion families in this way does not add new Yukawa or scalar couplings as the extra $(n_{\rm f} - 1)$ families only have gauge interactions. To define cross-family Yukawa couplings, one should follow the procedure in sec. 4.1.

Q.10. How do I compare with the literature?

Most outputs from DRalgo can directly be compared with the literature. For temporal scalars this requires some care. For example, if the original 4d theory has a SU(2) triplet, then two types of $A_0^2 \Sigma^2$ couplings are allowed:

$$\mathcal{L}_{\rm 3d} \supset \kappa_1 (A_0^a A_0^a)(\Sigma^a \Sigma^a) + \kappa_2 (A_0^a \Sigma^a)^2 \,. \tag{5.3}$$

DRalgo does not distinguish between these two terms since its output is stored in the form $\frac{1}{4!} A_0^a A_0^b \Sigma^c \Sigma^d \lambda_K^{abcd}$. To compare with the two parametrizations, the user can rewrite the κ_1/κ_2 basis in tensor form. To this end, it is necessary to additionally define the relevant terms of the effective model and compare to its Lagrangian. This procedure is analogous to creating the fundamental model where invariants can be compared using the command CompareInvariants[]. See the htm.m file for a worked example of the Higgs triplet model (HTM).

The matching of all possible allowed operators in the EFT is automatic in DRalgo. This way also previously disregarded effective coefficients can be determined. One example is a $\mathcal{L}_{\rm 3d} \supset \kappa \mathrm{Tr}\, C_0^3 B_0$ operator in the Standard Model. Here, C_0 is the gluon (temporal) field, and B_0 is the temporal hypercharge field. The output from DRalgo gives $\kappa \propto (2Y_q + Y_d + Y_u)$. See eq. (4.4) for the corresponding hypercharges and the 2hdm.m example file for the full expression

---

[11] This adds nF copies of all fermions.

Q.11. What is the functional basis used in the matching?

In addition to the variables `Lb` and `Lf` defined in eq. (2.37), two-loop diagrams also contain factors of

$$c = \frac{1}{2}\left(\ln\left(\frac{8\pi}{9}\right) + (\ln\zeta_2)' - 2\gamma_E\right),$$

(5.4)

where $\zeta_s = \zeta(s)$ is the Riemann zeta function and $(\ln\zeta_s)' = \zeta'(s)/\zeta(s)$. By default `DRalgo` uses the relations $\ln(2\pi) - (\ln\zeta_2)' = 1 - \gamma_E + (\ln\zeta_{-1})'$ and $1 + (\ln\zeta_{-1})' = 12\ln A$, where A is the Glaisher-Kinkelin constant. It is possible to convert the output of `DRalgo` to the conventions of [34] by using the replacement rule `Log[Glaisher]->-1/12(Lb+2cplus-EulerGamma)`, where `cplus=(c+Log[3T/`μ`])`. This rule is implemented as `PrintGenericBasis[]` in `DRalgo`.

Q.12. How do I define scalar cubic operators?

Scalar cubics are created analogously to quartics. Consider for example a SU(2) theory with a scalar doublet ϕ and a singlet S. We can then create the cubic operator $(\phi\phi^\dagger)S$ via

```
InputInv={{1,1,2},{True,False,True}}; (*ϕϕ†S*)
CubicTerm1=CreateInvariant[Group,RepScalar,InputInv][[1]]//Simplify;
VCubic=λC*CubicTerm1;
```

The corresponding tensor is defined with the command

```
λ3=GradCubic[VCubic];
```

Tadpoles are defined analogously. See the `2xsm.m` file for a worked example with two real scalars.

Q.13. Do I need to define tadpoles?

If tadpoles are allowed by symmetry we encourage the user to define them – even if they are absent at tree-level.

Q.14. How do I define Dirac and Majorana masses?

Fermion masses are defined analogously to scalar masses. We stress that `DRalgo` only works with Weyl fermions. For example, assume a theory with two Weyl fermions ψ_1 and ψ_2. The following terms are then possible

$$m_1\psi_1\psi_1 + m_2\psi_2\psi_2 + m_D\psi_1\psi_2 + \text{h.c.}$$

(5.5)

Where the first two terms are Majorana masses, and the third is a Dirac-type mass. These masses can be defined in `DRalgo` via

```
InputInv={{1,1},{True,True}}; (*ψ₁ψ₁*)
MassTerm1=CreateInvariantFermion[Group,RepFermion,InputInv][[1]]//Simplify;
InputInv={{2,2},{True,True}}; (*ψ₂ψ₂*)
MassTerm2=CreateInvariantFermion[Group,RepFermion,InputInv][[1]]//Simplify;
InputInv={{1,2},{True,True}}; (*ψ₁ψ₂*)
MassTerm3=CreateInvariantFermion[Group,RepFermion,InputInv][[1]]//Simplify;
```

The mass matrix is given by

```
FermionMasses=1/2*m1*MassTerm1+1/2*m2*MassTerm2+mD MassTerm3;
μIJ=GradMassFermion[FermionMasses];
μIJC=SparseArray[Simplify[Conjugate[μIJ]//Normal,Assumptions->{m1>0,m2>0,mD>0}]];
```

See the `WessZumino.m` notebook for a worked example.

Q.15. What if a model has $\overline{\Psi}_1\gamma_5\Psi_2$ terms?

Since `DRalgo` only uses Weyl fermions, the inclusion of other fermionic bilinear operators requires some extra work. One example is the operator $\overline{\Psi}_1\gamma_5\Psi_2$ which can be expanded in Weyl spinors

$$\Psi_1 = \begin{pmatrix} \psi_1^L \\ \psi_1^R \end{pmatrix}, \quad \Psi_2 = \begin{pmatrix} \psi_2^L \\ \psi_2^R \end{pmatrix},$$

(5.6)

such that

$$\overline{\Psi}_1\Psi_2 = \psi_1^R\psi_2^L + \psi_1^L\psi_2^R,$$

(5.7)

$$\overline{\Psi}_1\gamma_5\Psi_2 = -\psi_1^R\psi_2^L + \psi_1^L\psi_2^R.$$

(5.8)

These terms can then be implemented in `DRalgo` as explained in earlier examples.

Q.16. My mass matrix is not diagonal. Is the effective potential still calculable?

Yes, it is still possible to calculate the effective potential, just not in general. It is up to the user to diagonalize the mass matrix. But once done, the user has two choices. First, diagonalization-matrices can be given to DRalgo. Second, the user can do all the diagonalization themselves and reload the model. The first option is quick, but becomes protracted if the diagonalization is complicated or needs to be done perturbatively.

Consider for example the Standard Model. As before, field-dependent masses can be created via

```
DefineNewTensorsUS[μij,λ4,λ3,gvss,gvvv];
φvev={0,0,0,φ}//SparseArray;
DefineVEVS[φvev];
PrintTensorsVEV[];
```

To diagonalize the vector-bosons, we first need to extract the field-dependent mass-tensor:

```
MassMatrix=PrintTensorsVEV[];
VectorMass=MassMatrix[[2]]//Normal;
VectorEigenvectors=FullSimplify[
    Transpose[Normalize/@Eigenvectors[VectorMass[[11;;12,11;;12]]]],
    Assumptions->{g1>0,g2>0,φ>0}];
DVRot={{IdentityMatrix[10],0},{0,VectorEigenvectors}}//ArrayFlatten;
DSRot=IdentityMatrix[4];
RotateTensorsUSPostVEV[DSRot,DVRot];
```

After this the effective-potential calculation proceeds as before.

Q.17. Can DRalgo handle non-renormalizable operators?

Currently no. This feature will be implemented in future versions.

Q.18. How do I define complicated scalar potentials?

For most cases, scalar quartics can be easily defined using the examples above. For non-standard representations, the model-building tools in DRalgo might differ by a basis change from other conventions in the literature. Let us take an example, a SU(5) model with an adjoint scalar. Commonly this scalar is written as $\Phi = \Phi^a T^a$ where T^a are traceless hermitian matrices satisfying $\text{Tr}\, T^a T^b = \frac{1}{2}\delta^{ab}$. The most general (quartic) potential is

$$V_1 = \delta_1 (\text{Tr}\, \Phi^2)^2 + \delta_2 \text{Tr}\, \Phi^4 \,. \tag{5.9}$$

By default DRalgo uses GroupMath, which defines its invariants differently to those above. In fact, the two scalar quartic operators given in GroupMath can be linear combinations of those above. Thus, the result from DRalgo would be in the form

$$V_2 = \lambda_H \text{inv}_1(\Phi^4) + \lambda_S \text{inv}_2(\Phi^4) \,, \tag{5.10}$$

where the invariants $\text{inv}_{1,2}$ are the output from the CreateInvariant command. Fortunately, it is quite easy to find the relations between the V_1 and the V_2 basis. To do so, first, rewrite everything in tensor form

$$V_1 = 1/4! \lambda_{1,ijkl} \Phi_i \Phi_j \Phi_k \Phi_l \,, \tag{5.11}$$

and likewise for the DRalgo output

$$V_2 = 1/4! \lambda_{2,ijkl} \Phi_i \Phi_j \Phi_k \Phi_l \,. \tag{5.12}$$

Since λ_1 and λ_2 are defined in different bases, we want to compare invariants. Here we only need three:

$$\lambda_{ijkl}\delta_{ij}\delta_{kl} \,, \quad \lambda_{ijkl}\lambda_{ijkl} \,, \quad \lambda_{ijnm}\lambda_{nmkl}\lambda_{klij} \,. \tag{5.13}$$

Comparing the invariants one finds

$$\lambda_H = \frac{1}{960}\sqrt{\frac{23}{14}}(130\delta_1 + 47\delta_2) \,, \quad \lambda_S = -\frac{5}{64}\sqrt{\frac{13}{42}}(2\delta_1 - \delta_2) \,. \tag{5.14}$$

The above procedure works for any representation as long as the user can write the quartic operator from DRalgo in their preferred form. See SU5.m for a concrete example.

Q.19. Can I use dimensionally reduced theory for calculating nucleation rates?

Certainly. The user can use the effective theory and directly find the bounce in the 3d. To first approximation the nucleation rate is $e^{-S_{3d}}$, where S_{3d} is the bounce action in the dimensionally reduced theory. See [85] for the original work, and [76,86–88] for recent examples.

Q.20. Can I study GUT models with DRalgo?

Most GUT models can be readily studied with DRalgo. However, for models with large representations the running-time rises rather rapidly. The bottleneck is not the complexity of e.g. the scalar potential but rather the size of the representations of particles. For example, DRalgo rapidly handles the general Pati-Salam, SU(5), 3HDM, or left-right symmetric model – even with 20–30 free parameters in the potential. See SU5.m, 3hdm.m, and LRSymmetric.m for explicit implementations.

Q.21. How much RAM does `DRalgo` require?

The required RAM is negligible for most models. However, for models with 50–100 dimensional representations, the requirements start to rise. And for said scenario around 1 GB of RAM is required. If the package drains too much RAM, it is recommended to not include two-loop contributions.

Q.22. How should I report possible bugs?

We are grateful if any bugs are reported. We kindly ask users to supply both a short description and a `Mathematica` file describing the error. These reports and attached files can be provided via the Issue Tracker on https://github.com/DR-algo/DRalgo.

6. Outlook

High-temperature field theory is pestered by large radiative corrections, which compromises perturbative calculations. In effect, perturbation theory needs to be reorganized in terms of thermal resummations. While at leading order only thermally corrected masses are required, new contributions to couplings arise at higher orders. In addition, there are currently several different schemes for incorporating thermal masses – all depending substantially on the renormalization scale.

As an effective field theory, dimensional reduction by-passes these issues. The ultraviolet sector of the theory is controlled by a matching computation and the infrared sector is resummed to all orders. Indeed, as discussed in this paper, using a three dimensional EFT unambiguously resums masses and couplings. Using the framework is simple from a practical standpoint since three-dimensional theories are super renormalizable. Concretely, only mass parameters are RG-scale dependent and their beta functions are exact at two-loop level. This property not only allows for a straightforward perturbative treatment, but also provides an attractive framework to a non-perturbative lattice study of the 3d EFT since in the effective theory relations between continuum and lattice parameters are exact at two-loop level [89].

Hitherto, dimensional reduction has been used sparsely. With this paper and the associated software, we aim to change this by automating the EFT construction. In summary, `DRalgo` calculates all effective couplings and masses in the effective theory, the leading-order beta functions both in the 4d parent and 3d effective theory, as well as the effective potential within the EFT. This facilitates studying models with dimensional reduction and requires merely three-dimensional calculations that are analogous to zero-temperature computations. Since the effective theory is fully bosonic, the perturbative calculation can be compared with lattice simulations. This has one clear benefit as a large number of fundamental 4d theories can map into the same effective theory given the mass hierarchy of the additional scalars. Not only does the lattice provide an invaluable cross-check, but pre-existing simulations can be reused and applied to new BSM theories.

In conclusion, gravitational waves have opened up a new gateway to the early universe, and particle physics stands at its threshold. Upcoming experiments are fast approaching both at future particle colliders and gravitational wave observatories. It is therefore important to control theoretical uncertainties at unprecedented precision. In this venture, dimensional reduction is the tool of choice, and `DRalgo` its harbinger.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Appendix A. Interfacing `DRalgo` output

In this appendix, we collect additional material that is not part of the package itself.

A.1. From 3d effective theory to thermodynamics

A possible interface between the output of `DRalgo`, in the 3-dimensional theory, and the determination of thermodynamic quantities is implemented in `./examples/ah-thermo.m`. This implementation comes with a disclaimer: we present a simplified algorithm suitable for the simple case of the Abelian-Higgs model (cf. eq. (3.1)) but this is not part of the package itself. We encourage users to implement their own numerical minimisation routines for thermodynamics optimized for individual models. The algorithm given here is not gauge invariant and should be used with discretion.

The Algorithm 2 illustrates our implementation. In the simplified algorithm `ah-thermo.m` the loop over 4-dimensional variables c_i

Algorithm 2 The `DRalgo` algorithm output is interfaced in `./examples/ah-thermo.m` with functions `solveBetas[]`, `DRstep[]` and `Veff3d[]`, which are called by `findThermo[]`.

Input: Four-dimensional theory parameters $c_i \in \{c_1, \ldots, c_n\}$ and scale factor X
for all $c_i \in \{c_1, \ldots, c_n\}$ **do**
    Call `findThermo[X, c_i]` {
        Call $c_i(\bar{\mu})$ = `solveBetas[c_i]`
        for all $T_i \in \{T_{\min}, \ldots, T_{\max}\}$ **do**   // e.g. binary search
            Fix T-dependent RG-scale: $\bar{\mu} = X\pi T$
            Call $c_{i,3d}$ = `DRstep[T, \bar{\mu}, c_i]`
            Minimize the effective potential by `NMinimize[Re[Veff3d[φ, c_{i,3d}]]]`
        end for
        Return $\{T_c, \phi_c/T_c, L/T_c^4\}$, based on degenerate minima.
    }
end for
Output: Thermodynamics as function of $\{c_1, \ldots, c_n\}$
Export the data
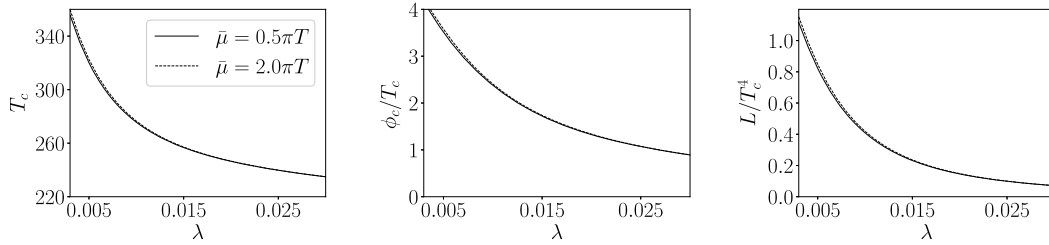Plot by Python: `.examples/ah-thermo-python-plots/plot.py`

Fig. 4. Thermodynamics of the Abelian-Higgs model as function of λ (cf. eq. (3.1)) at fixed $g^2 = 0.42$ and $M = 100$ as obtained in `./examples/ah-thermo.m`. Both M and T are in arbitrary units of mass. The RG-scale is varied at the edges of the interval $\bar{\mu} = (0.5 \ldots 2.0)\pi T$. While the curves are barely discernible, this showcases that the scale-dependence is indeed a higher-order effect once at full NLO dimensional reduction. The additional RG-scale of the 3d EFT is not varied in the plots.

contains only a single λ and for simplicity and faster running time $M = 100$ and $g^2 = 0.42$ are fixed. Here, M is the Higgs mass in arbitrary units related to the $\overline{\text{MS}}$ mass parameter by the tree-level relation $\mu^2 = -M^2/2$. The critical temperature T_c is defined from the condition that the effective potential at the broken minimum vanishes since the potential at the symmetric minimum at the origin is normalized to zero. To find this condition, we use an elementary binary search.

As an indicator of the transition strength, we also output the value of the background field at the critical temperature, ϕ_c/T_c, that describes the discontinuity of the minima at T_c. Since ϕ_c/T_c is gauge-dependent, it should not be given physical meaning but used as a rough indicator that correlates positively with the phase transition strength. This strength can be defined in terms of the released latent heat at the critical temperature T_c

$$L = T\Delta p' = T\Delta \frac{dV_{\text{eff}}^{4d}}{dT} = T^2 \Delta \frac{dV_{\text{eff}}^{3d}}{dT} , \tag{A.1}$$

where p' is the derivative of the pressure with respect to temperature and $\Delta \equiv (\ldots)_{\text{low-}T} - (\ldots)_{\text{high-}T}$ denotes the difference between the broken and symmetric phase. The T-derivative is approximated as a finite difference $dT = 0.1$. The symmetric-phase part does not contribute here since we normalized the effective potential to zero at all T at the origin. For a commented documentation of the technicalities of this implementation see directly the source `ah-thermo.m`.

The described perturbative determination is done in Landau gauge with $\xi = 0$ in eq. (2.19) and the results for T_c and L/T_c^4 are gauge-dependent. Despite this simple user-friendly example, a recipe for a more sophisticated, gauge-invariant determination can be found in e.g. [90] (cf. refs. therein). The final output data for thermodynamics is
stored in `./examples/ah-thermo-python-plots/*.dat`,
plotted in `./examples/ah-thermo-python-plots/plot.py`, and shown in Fig. 4.

We encourage `DRalgo` users to develop and optimize their individual implementations for algorithms to determine thermodynamic properties. While including efficient algorithms for determining thermodynamics is conceivable for future versions of `DRalgo`, in the current version `1.0` these features are not implemented in the package itself.

References

[1] L. Canetti, M. Drewes, M. Shaposhnikov, New J. Phys. 14 (2012) 095012, arXiv:1204.4186.
[2] G. Elor, et al., in: 2022 Snowmass Summer Study, vol. 3, 2022, arXiv:2203.05010.
[3] M.E. Shaposhnikov, Nucl. Phys. B 287 (1987) 757.
[4] M.E. Shaposhnikov, JETP Lett. 44 (1986) 465.
[5] V.A. Kuzmin, V.A. Rubakov, M.E. Shaposhnikov, Phys. Lett. B 155 (1985) 36.
[6] A.D. Sakharov, Pis'ma Zh. Eksp. Teor. Fiz. 5 (1967) 32.
[7] K. Kajantie, M. Laine, K. Rummukainen, M.E. Shaposhnikov, Nucl. Phys. B 466 (1996) 189, arXiv:hep-lat/9510020.
[8] M. Gürtler, E.-M. Ilgenfritz, A. Schiller, Phys. Rev. D 56 (1997) 3888, arXiv:hep-lat/9704013.
[9] K. Kajantie, M. Laine, K. Rummukainen, M.E. Shaposhnikov, Phys. Rev. Lett. 77 (1996) 2887, arXiv:hep-ph/9605288.
[10] M.J. Ramsey-Musolf, J. High Energy Phys. 09 (2020) 179, arXiv:1912.07189.
[11] L.S. Friedrich, M.J. Ramsey-Musolf, T.V.I. Tenkanen, V.Q. Tran, Addressing the gravitational wave - collider inverse problem, arXiv:2203.05889.
[12] LISA Collaboration, P. Amaro-Seoane, et al., Laser interferometer space antenna, arXiv:1702.00786.
[13] S. Kawamura, et al., Class. Quantum Gravity 23 (2006) S125.

[14] W.-H. Ruan, Z.-K. Guo, R.-G. Cai, Y.-Z. Zhang, Int. J. Mod. Phys. A 35 (2020) 2050075, arXiv:1807.09495.
[15] G.M. Harry, P. Fritschel, D.A. Shaddock, W. Folkner, E.S. Phinney, Class. Quantum Gravity 23 (2006) 4887.
[16] J. de Vries, M. Postma, J. van de Vis, G. White, J. High Energy Phys. 01 (2018) 089, arXiv:1710.04061.
[17] G.C. Dorsch, S.J. Huber, T. Konstandin, J. Cosmol. Astropart. Phys. 12 (2018) 034, arXiv:1809.04907.
[18] E. Hall, T. Konstandin, R. McGehee, H. Murayama, G. Servant, J. High Energy Phys. 04 (2020) 042, arXiv:1910.08068.
[19] I. Baldes, G. Servant, J. High Energy Phys. 10 (2018) 053, arXiv:1807.08770.
[20] M. Chala, M. Ramos, M. Spannowsky, Eur. Phys. J. C 79 (2019) 156, arXiv:1812.01901.
[21] D. Croon, O. Gould, P. Schicho, T.V.I. Tenkanen, G. White, J. High Energy Phys. 04 (2021) 055, arXiv:2009.10080.
[22] O. Gould, J. High Energy Phys. 04 (2021) 057, arXiv:2101.05528.
[23] A. Alves, T. Ghosh, H.-K. Guo, K. Sinha, D. Vagie, J. High Energy Phys. 04 (2019) 052, arXiv:1812.09333.
[24] L. Niemi, P. Schicho, T.V.I. Tenkanen, Phys. Rev. D 103 (2021) 115035, arXiv:2103.07467.
[25] N.F. Bell, M.J. Dolan, L.S. Friedrich, M.J. Ramsey-Musolf, R.R. Volkas, J. High Energy Phys. 21 (2020) 098, arXiv:2010.13376.
[26] L. Niemi, M.J. Ramsey-Musolf, T.V.I. Tenkanen, D.J. Weir, Phys. Rev. Lett. 126 (2021) 171802, arXiv:2005.11332.
[27] A.D. Linde, Phys. Lett. B 96 (1980) 289.
[28] E. Braaten, Phys. Rev. Lett. 74 (1995) 2164, arXiv:hep-ph/9409434.
[29] P.B. Arnold, O. Espinosa, Phys. Rev. D 47 (1993) 3546, arXiv:hep-ph/9212235.
[30] K. Farakos, K. Kajantie, K. Rummukainen, M.E. Shaposhnikov, Nucl. Phys. B 425 (1994) 67, arXiv:hep-ph/9404201.
[31] O. Gould, T.V.I. Tenkanen, J. High Energy Phys. 06 (2021) 069, arXiv:2104.04399.
[32] P.H. Ginsparg, Nucl. Phys. B 170 (1980) 388.
[33] T. Appelquist, R.D. Pisarski, Phys. Rev. D 23 (1981) 2305.
[34] K. Kajantie, M. Laine, K. Rummukainen, M.E. Shaposhnikov, Nucl. Phys. B 458 (1996) 90, arXiv:hep-ph/9508379.
[35] E. Braaten, A. Nieto, Phys. Rev. D 51 (1995) 6990, arXiv:hep-ph/9501375.
[36] K. Kajantie, M. Karjalainen, M. Laine, J. Peisa, Nucl. Phys. B 520 (1998) 345, arXiv:hep-lat/9711048.
[37] J.O. Andersen, Phys. Rev. D 59 (1999) 065015, arXiv:hep-ph/9709418.
[38] J.O. Andersen, Z. Phys. C 75 (1997) 147, arXiv:hep-ph/9606357.
[39] K. Farakos, K. Kajantie, K. Rummukainen, M.E. Shaposhnikov, Nucl. Phys. B 442 (1995) 317, arXiv:hep-lat/9412091.
[40] J.O. Andersen, T. Gorda, A. Helset, et al., Phys. Rev. Lett. 121 (2018) 191802, arXiv:1711.09849.
[41] L. Niemi, H.H. Patel, M.J. Ramsey-Musolf, T.V.I. Tenkanen, D.J. Weir, Phys. Rev. D 100 (2019) 035002, arXiv:1802.10500.
[42] J. Hirvonen, Intuitive method for constructing effective field theories, arXiv:2205.02687.
[43] A. Rajantie, Nucl. Phys. B 501 (1997) 521, arXiv:hep-ph/9702255.
[44] J.O. Andersen, N. Haque, M.G. Mustafa, M. Strickland, Phys. Rev. D 93 (2016) 054045, arXiv:1511.04660.
[45] J.O. Andersen, E. Braaten, E. Petitgirard, M. Strickland, Phys. Rev. D 66 (2002) 085016, arXiv:hep-ph/0205085.
[46] S. Baum, M. Carena, N.R. Shah, C.E.M. Wagner, Y. Wang, J. High Energy Phys. 03 (2021) 055, arXiv:2009.10743.
[47] I.P. Ivanov, C.C. Nishi, J. High Energy Phys. 01 (2015) 021, arXiv:1410.6139.
[48] P.S. Bhupal Dev, R.N. Mohapatra, W. Rodejohann, X.-J. Xu, J. High Energy Phys. 02 (2019) 154, arXiv:1811.06869.
[49] T. Appelquist, B.A. Dobrescu, A.R. Hopper, Phys. Rev. D 68 (2003) 035012, arXiv:hep-ph/0212073.
[50] P.M. Schicho, T.V.I. Tenkanen, J. Österman, J. High Energy Phys. 06 (2021) 130, arXiv:2102.11145.
[51] D. Bodeker, Nucl. Phys. B 559 (1999) 502, arXiv:hep-ph/9905239.
[52] G.D. Moore, K. Rummukainen, A. Tranberg, J. High Energy Phys. 04 (2001) 017, arXiv:hep-lat/0103036.
[53] G.D. Moore, K. Rummukainen, Phys. Rev. D 63 (2001) 045002, arXiv:hep-ph/0009132.
[54] M. Laine, Y. Schröder, Phys. Rev. D 73 (2006) 085009, arXiv:hep-ph/0603048.
[55] J. Ghiglieri, A. Kurkela, M. Strickland, A. Vuorinen, Phys. Rep. 880 (2020) 1, arXiv:2002.10188.
[56] A.K. Rajantie, Nucl. Phys. B 480 (1996) 729, arXiv:hep-ph/9606216.
[57] J. Möller, Y. Schröder, J. High Energy Phys. 08 (2012) 025, arXiv:1207.1309.
[58] M. Laine, P. Schicho, Y. Schröder, J. High Energy Phys. 05 (2018) 037, arXiv:1803.08689.
[59] J.M. Overduin, P.S. Wesson, Phys. Rep. 283 (1997) 303, arXiv:gr-qc/9805018.
[60] T. Cohen, PoS TASI2018 (2019) 011, arXiv:1903.03622.
[61] T. Matsubara, Prog. Theor. Phys. 14 (1955) 351.
[62] S.P. Martin, Phys. Rev. D 96 (2017) 096005, arXiv:1709.02397.
[63] S.P. Martin, H.H. Patel, Phys. Rev. D 98 (2018) 076008, arXiv:1808.07615.
[64] M.E. Machacek, M.T. Vaughn, Nucl. Phys. B 249 (1985) 70.
[65] M.E. Machacek, M.T. Vaughn, Nucl. Phys. B 236 (1984) 221.
[66] M.E. Machacek, M.T. Vaughn, Nucl. Phys. B 222 (1983) 83.
[67] V. Shtabovenko, R. Mertig, F. Orellana, Comput. Phys. Commun. 207 (2016) 432, arXiv:1601.01167.
[68] V. Shtabovenko, R. Mertig, F. Orellana, Comput. Phys. Commun. 256 (2020) 107478, arXiv:2001.04407.
[69] B. Ruijl, T. Ueda, J. Vermaseren, FORM version 4.2, arXiv:1707.06453.
[70] H.K. Dreiner, H.E. Haber, S.P. Martin, Phys. Rep. 494 (2010) 1, arXiv:0812.1594.
[71] E. Braaten, A. Nieto, Phys. Rev. D 53 (1996) 3421, arXiv:hep-ph/9510408.
[72] K.G. Chetyrkin, F.V. Tkachov, Nucl. Phys. B 192 (1981) 159.
[73] F.V. Tkachov, Phys. Lett. B 100 (1981) 65.
[74] M. Nishimura, Y. Schröder, J. High Energy Phys. 09 (2012) 051, arXiv:1207.4042.
[75] R.M. Fonseca, Comput. Phys. Commun. 267 (2021) 108085, arXiv:2011.01764.
[76] J. Hirvonen, J. Löfgren, M.J. Ramsey-Musolf, P. Schicho, T.V.I. Tenkanen, J. High Energy Phys. 07 (2022) 135, arXiv:2112.08912.
[77] M. Karjalainen, J. Peisa, Z. Phys. C 76 (1997) 319, arXiv:hep-lat/9607023.
[78] G.C. Branco, P.M. Ferreira, L. Lavoura, M.N. Rebelo, M. Sher, J.P. Silva, Phys. Rep. 516 (2012) 1, arXiv:1106.0034.
[79] J.F. Gunion, H.E. Haber, G.L. Kane, S. Dawson, The Higgs Hunter's Guide, vol. 80, 2000.
[80] H. Georgi, Lie Algebras in Particle Physics, vol. 54, 2nd ed., Perseus Books, Reading, MA, 1999.
[81] R. Slansky, Phys. Rep. 79 (1981) 1.
[82] M. Losada, Phys. Rev. D 56 (1997) 2893, arXiv:hep-ph/9605266.
[83] J.O. Andersen, Eur. Phys. J. C 11 (1999) 563, arXiv:hep-ph/9804280.
[84] T. Gorda, A. Helset, L. Niemi, T.V.I. Tenkanen, D.J. Weir, J. High Energy Phys. 02 (2019) 081, arXiv:1802.05056.
[85] A.D. Linde, Nucl. Phys. B 216 (1983) 421.
[86] O. Gould, J. Hirvonen, Phys. Rev. D 104 (2021) 096015, arXiv:2108.04377.
[87] A. Ekstedt, Eur. Phys. J. C 82 (2022) 173, arXiv:2104.11804.
[88] A. Ekstedt, J. High Energy Phys. 08 (2022) 115, arXiv:2201.07331.
[89] M. Laine, A. Rajantie, Nucl. Phys. B 513 (1998) 471, arXiv:hep-lat/9705003.
[90] P. Schicho, T.V.I. Tenkanen, G. White, J. High Energy Phys. 11 (2022) 047, arXiv:2203.04284.