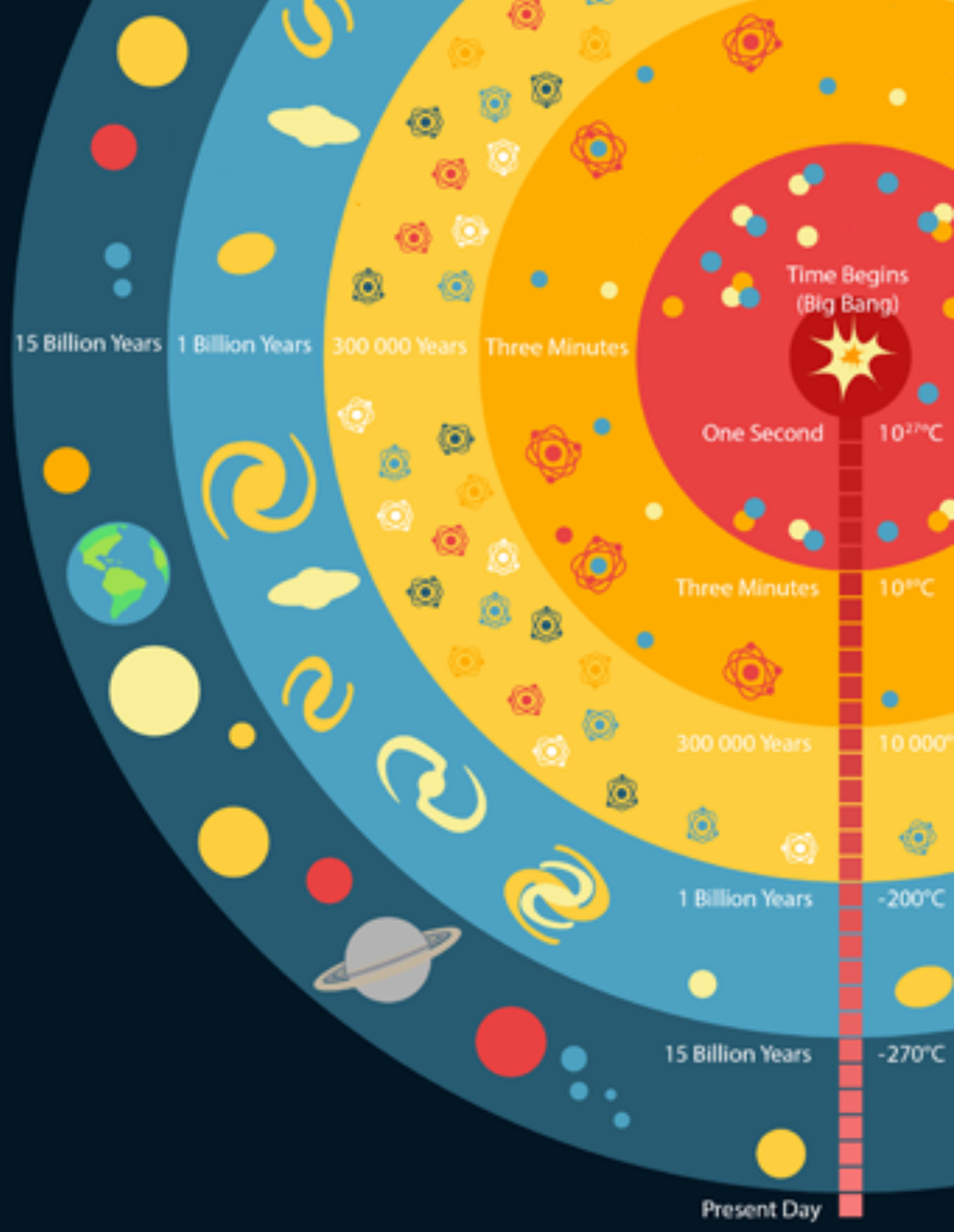
第一期新物理冬季学校WINP2025

电弱相变

河南师范大学 理论物理中心

张阳



电弱相变

1、~~动机、概念、前沿~~（参考边立功老师课程）

2、希格斯有效势的定义

3、相变参数的定义

4、相变计算工具

5、模型采样工具

希格斯有效势

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

➤ 有效势的定义

热力学与统计物理

系统的基态：吉布斯自由能的极小值

有外加磁场H下的配分函数（对系统所有可能的态求和）：

$$Z(H) = e^{-\beta F(H)} = \int \mathscr{D}s e^{-\beta \int \mathrm{d}x\, [\mathscr{H}(s) - Hs(x)]}$$

其中 $\beta = \dfrac{1}{k_B T}$，k_B 是波尔兹曼常数，T 是系统的绝对温度，

F 亥姆霍兹自由能：

$$F = -\frac{1}{\beta} \ln Z$$

通过勒让德变换，得到吉布斯自由能：

$$G = F - H \left.\frac{\partial F}{\partial J}\right|_{\beta \text{ fixed}}$$

量子场论

系统的基态（真空）：有效势的极小值

标量场ϕ的生成泛函（对系统所有可能的路径求和）：

$$Z[J] = e^{-iW[J]} = \int \mathscr{D}\phi e^{i \int \mathrm{d}x [L - J\phi]}$$

连通生成泛函：

$$W[J] = -i \ln Z[J]$$

通过勒让德变换，得到有效作用量：

$$\Gamma = W[J] - \int d^4 y J(y) \frac{\delta W[J]}{\delta J(y)}$$

可以提取出时空部分，定义有效势

$$\Gamma[\phi] = \int d^4 x [-V_{\text{eff}}(\phi) + \frac{1}{2}(\partial_\mu \phi)^2 Z(\phi) + \cdots$$

希格斯有效势

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

- 有效势的定义

Magnetic System	Quantum Field Theory
$\mathbf{x}$	$x = (t, \mathbf{x})$
$s(\mathbf{x})$	$\phi(x)$
H	$J(x)$
$\mathcal{H}(s)$	$\mathcal{L}(\phi)$
$Z(H)$	$Z[J]$
$F(H)$	$E[J]$
M	$\phi_{\rm cl}(x)$
$G(M)$	$-\Gamma[\phi_{\rm cl}]$

See Peskin&Schroeder's testbook, Section 11.3

$$Z[J] = \sum_{n=0}^{\infty} \frac{(i)^n}{n!} \int dx_1 \cdots dx_n G_n x_1, \cdots, x_n J(x_1) \cdots J(x_n)$$

$$iW[J] = \sum_{n=0}^{\infty} \frac{(i)^n}{n!} \int dx_1 \cdots dx_n G_n^C x_1, \cdots, x_n J(x_1) \cdots J(x_n)$$

$$\Gamma[\phi_{\rm cl}] = \sum_{n=0}^{\infty} \frac{(i)^n}{n!} \int dx_1 \cdots dx_n G_n^{\rm 1P1} x_1, \cdots, x_n \phi_(x_1) \cdots \phi_(x_n)$$

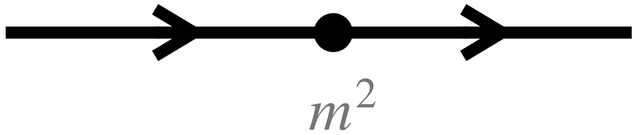
$$V_{\rm eff} = -\sum_{n=0}^{\infty} \frac{\phi^n}{n!} G_n^{\rm 1P1}(p_i = 0)$$

希格斯有效势

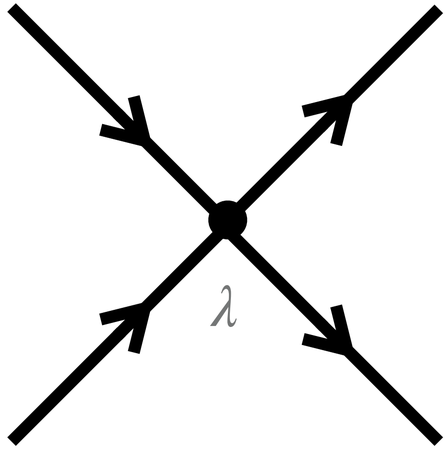
河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

- 有效势的定义

$$V_{\text{eff}} = -\sum_{n=0}^{\infty} \frac{\phi^n}{n!} G_n^{1\text{P}1}(p_i = 0)$$



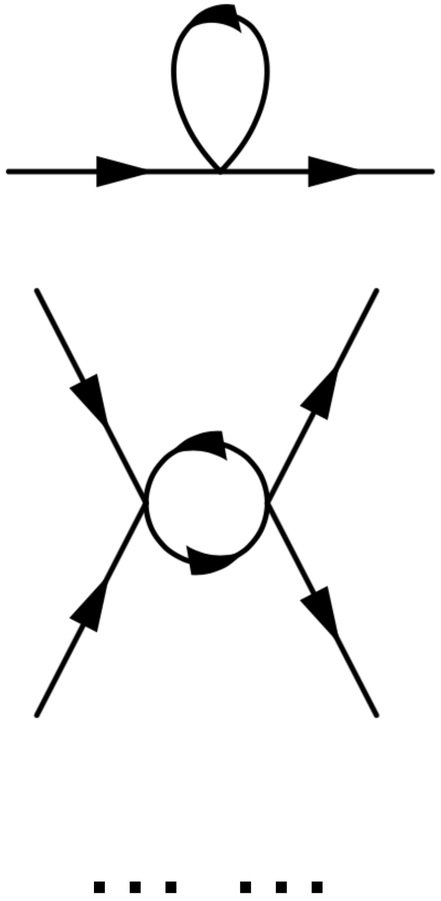
- $V_0 = \frac{1}{2} m^2 \phi^2 + \frac{1}{4!} \lambda \phi^4$



希格斯有效势

- 有效势的定义

$$V_{\text{eff}} = -\sum_{n=0}^{\infty} \frac{\phi^n}{n!} G_n^{1\text{P}1}(p_i = 0)$$



- $V_0 = \frac{1}{2} m^2 \phi^2 + \frac{1}{4!} \lambda \phi^4$

- $V_1 = i \sum_{n=1}^{\infty} \int \frac{d^4 p}{(2\pi)^4} \frac{1}{2n} \left[\frac{i}{p^2 - m^2 + i\epsilon} \right]^n \left[\frac{-i\lambda}{2} \right]^n \phi^{2n}$

希格斯有效势

- 将其转化为对数函数，$\ln(1-x) = -\sum_{n=1}^{\infty} x^n/n$

$$V_1 = i\sum_{n=1}^{\infty}\int\frac{d^4p}{(2\pi)^4}\frac{1}{2n}\left[\frac{\frac{1}{2}\lambda\phi^2}{p^2-m^2+i\epsilon}\right]^n = -\frac{i}{2}\int\frac{d^4p}{(2\pi)^4}\ln\left[1-\frac{\frac{1}{2}\lambda\phi^2}{p^2-m^2+i\epsilon}\right]$$

Wick 转动 $\bar{p}_0 = ip_0$ ；

$$= \frac{1}{2}\int\frac{d^4p}{(2\pi)^4}\ln\left[1+\frac{\frac{1}{2}\lambda\phi^2}{p^2+m^2+i\epsilon}\right] = \frac{1}{2}\int\frac{d^4p}{(2\pi)^4}\ln\left[\frac{p^2+m^2+\frac{1}{2}\lambda\phi^2}{p^2+m^2+i\epsilon}\right]$$

- $$= \frac{1}{2}\left\{\int\frac{d^4p}{(2\pi)^4}\ln\left[p^2+m^2+\frac{1}{2}\lambda\phi^2\right] - \int\frac{d^4p}{(2\pi)^4}\ln\left[p^2+m^2+i\epsilon\right]\right\},$$

- 丢到与ϕ无关的项，得到$V_1 = -\frac{i}{2}\int\frac{d^4p}{(2\pi)^4}\ln\left[p^2+m^2(\phi)\right]$，其中 $m^2(\phi) = m^2+\frac{1}{2}\lambda\phi^2$

希格斯有效势

- 维数正规化；利用$\ln x = \int dx \frac{1}{x}$,

$$V_1 = \frac{1}{2}\int \frac{d^4p}{(2\pi)^4}\ln[p^2+m^2(\phi)] = (\mu^2)^{2-d/2}\frac{1}{2}\int\frac{d^dp}{(2\pi)^d}\ln[p^2+m^2(\phi)]$$

$$= \int dm^2(\phi)\frac{(\mu^2)^{2-d/2}}{2}\int\frac{d^dp}{(2\pi)^d}\frac{1}{p^2+m^2(\phi)}$$

- 利用积分公式$\int\frac{d^dp}{(2\pi)^d}\frac{1}{(p^2-m^2)^n} = \frac{1}{(4\pi)^{d/2}}\frac{\Gamma(n-d/2)}{\Gamma(n)}\left(\frac{1}{m^2}\right)^{n-d/2}$得到

$$V_1(\phi) = \int dm^2(\phi)\left\{\frac{(\mu^2)^{2-d/2}}{2}\left(\frac{1}{m^2}\right)^{1-d/2}\frac{1}{(4\pi)^{d/2}}\frac{\Gamma(1-d/2)}{\Gamma(1)}\right\} = \frac{(\mu^2)^{2-d/2}}{2}\frac{1}{(\frac{d}{2})}\left(\frac{1}{m^2}\right)^{-d/2}\frac{1}{(4\pi)^{d/2}}\frac{1}{(1-\frac{d}{2})}\Gamma(2-d/2)$$

$$= \frac{m^4}{2}\frac{1}{\frac{d}{2}(1-\frac{d}{2})}\frac{1}{(4\pi)^{d/2}}\left(\frac{\mu^2}{m^2}\right)^{2-\frac{d}{2}}\Gamma(2-d/2)$$

希格斯有效势

- 取 $d \to 4$，得到

$$V_1(\phi) = \frac{m^4}{2}\frac{1}{\frac{d}{2}(1-\frac{d}{2})}\frac{1}{(4\pi)^{d/2}}\left(\frac{\mu^2}{m^2}\right)^{2-\frac{d}{2}}\Gamma(2-d/2) \to \frac{m^4}{2}\frac{1}{-2}\frac{1}{(4\pi)^2}\left(\frac{2}{\epsilon}-\gamma_E+\frac{3}{2}+\ln(4\pi)-\ln(\frac{m^2}{\mu^2})\right)$$

- 采用$\overline{\text{MS}}$方案，丢掉发散项，得到

$$V_1(\phi) = -\frac{m^4}{2}\frac{1}{2}\frac{1}{(4\pi)^2}\left[-\ln(\frac{m^2}{\mu^2})+\frac{3}{2}\right] = \frac{m^4}{64\pi^2}\left[\ln(\frac{m^2}{\mu^2})-\frac{3}{2}\right]$$

➤ 零温有效势：

- $$V(\phi) = V_0(\phi)+V_1(\phi) = \frac{1}{2}m^2\phi^2+\frac{1}{4!}\lambda\phi^4+\frac{m(\phi)^4}{64\pi^2}\left[\ln(\frac{m(\phi)^2}{\mu^2})-\frac{3}{2}\right]$$

希格斯有效势

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

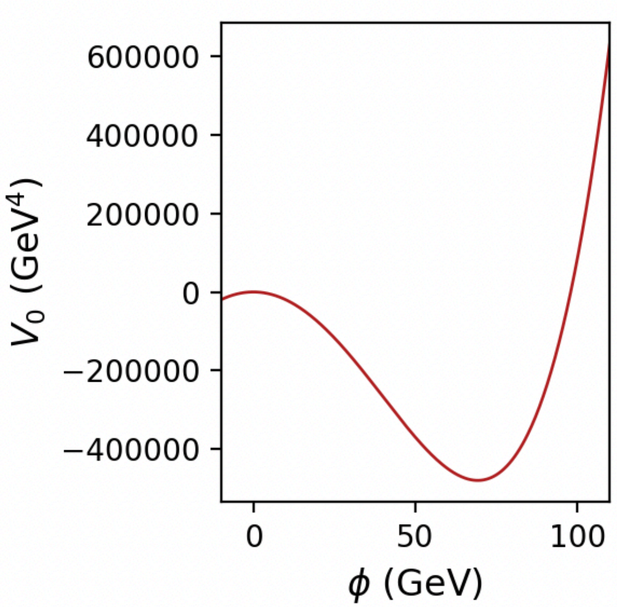
- 零温有效势：

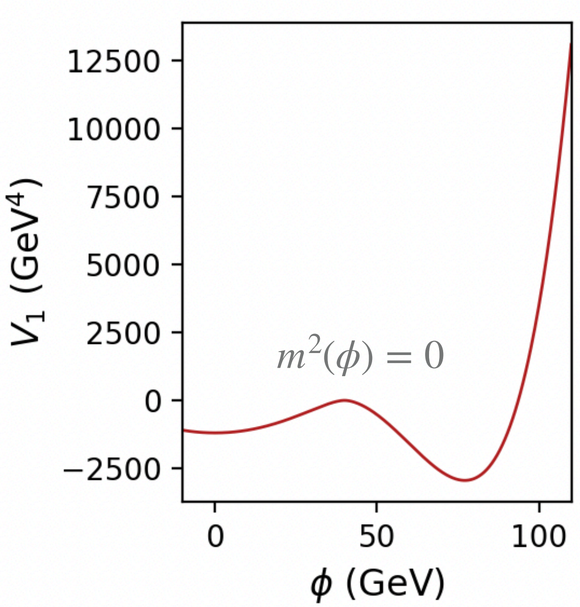
$$m^2(\phi) = m^2 + \frac{1}{2}\lambda\phi^2$$

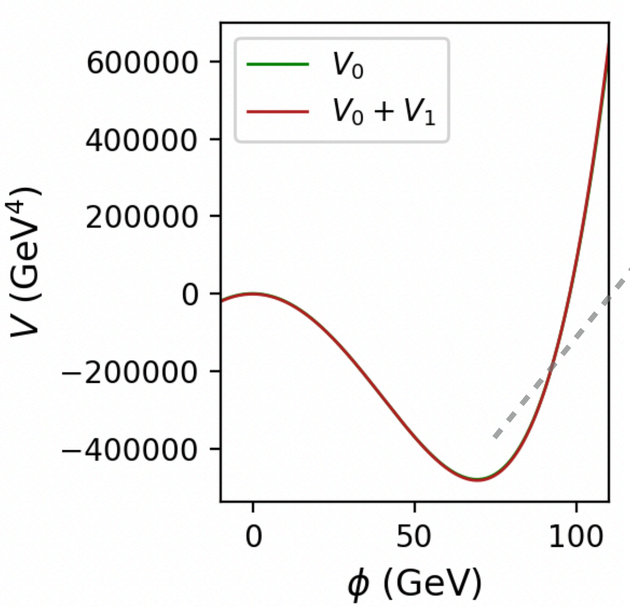
取实部

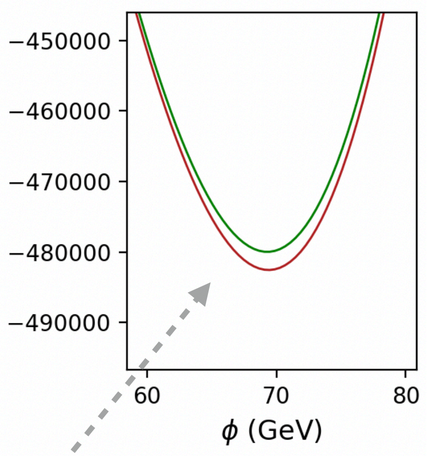
$$V(\phi) = V_0(\phi) + V_1(\phi) = \frac{1}{2}m^2\phi^2 + \frac{1}{4!}\lambda\phi^4 + \frac{m(\phi)^4}{64\pi^2}\left[\ln\left(\frac{m(\phi)^2}{\mu^2}\right) - \frac{3}{2}\right]$$

$m^2 = -20^2, \lambda = 0.5$









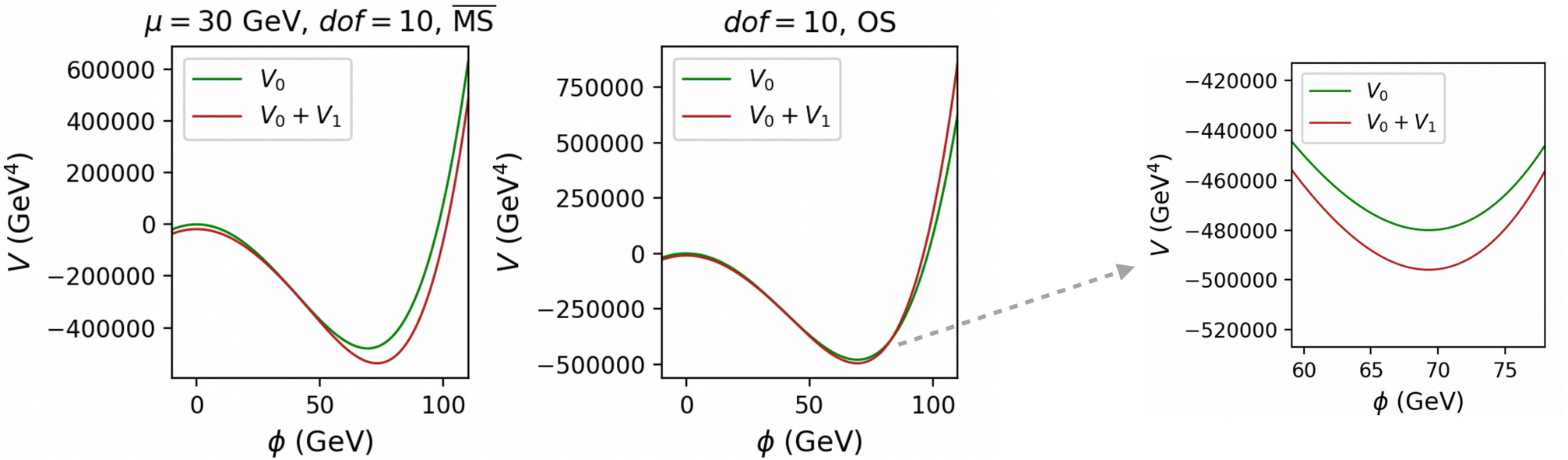
希格斯有效势

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

- 修改的最小剪除方案 VS 在壳方案

  - 在壳方案: $V(\phi) = V_0(\phi) + V_1(\phi) + \delta V(\phi), \quad \frac{\partial V}{\partial \phi} = \frac{\partial V_0}{\partial \phi} = 0, \quad \frac{\partial^2 V}{\partial \phi^2} = \frac{\partial^2 V_0}{\partial \phi^2} = \hat{m}^2$

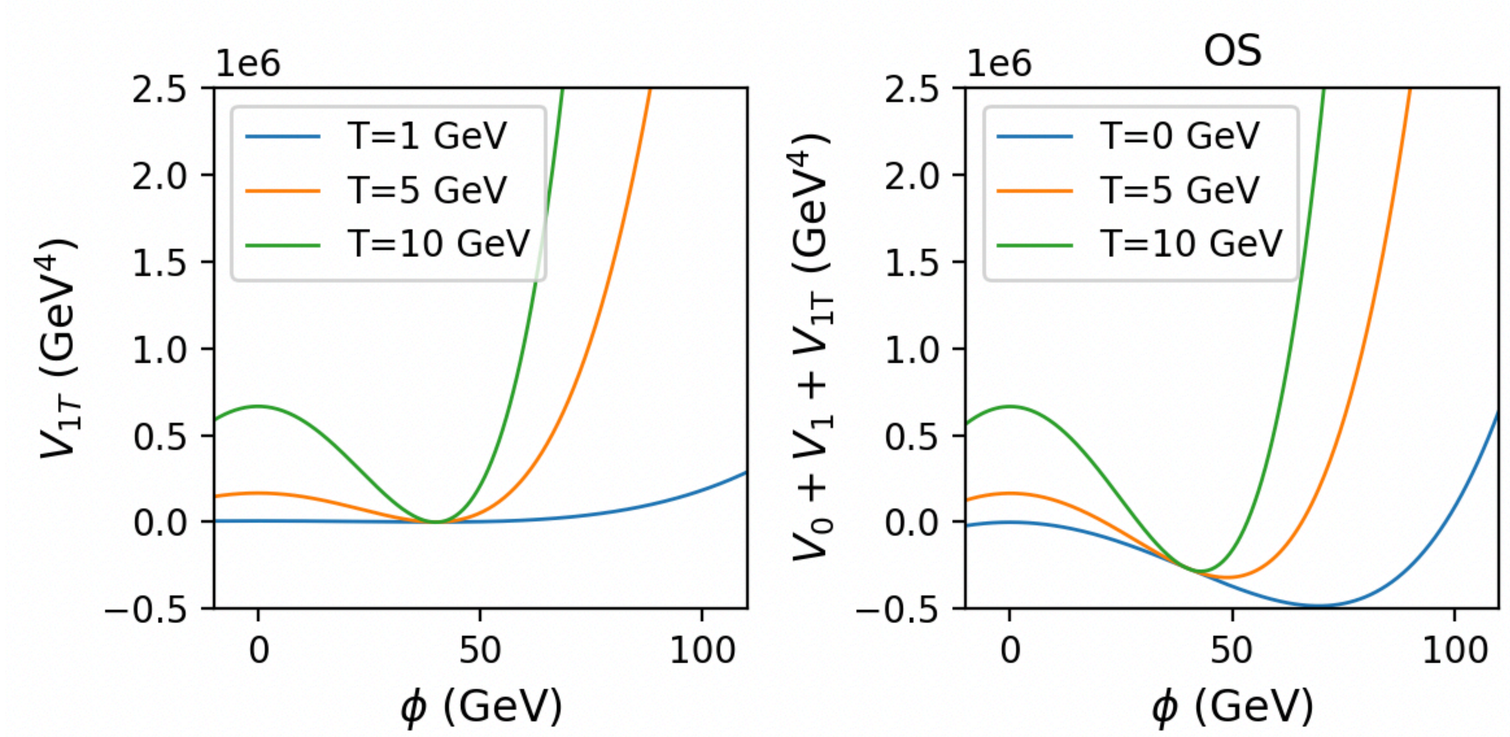
  - 在一些模型中， $V_1(\phi) + \delta V(\phi) = \frac{m(\phi)^4}{64\pi^2}\left[\ln(\frac{m(\phi)^2}{\hat{m}^2}) - \frac{1}{2}\right], \quad \hat{m} = m(\phi = v)$

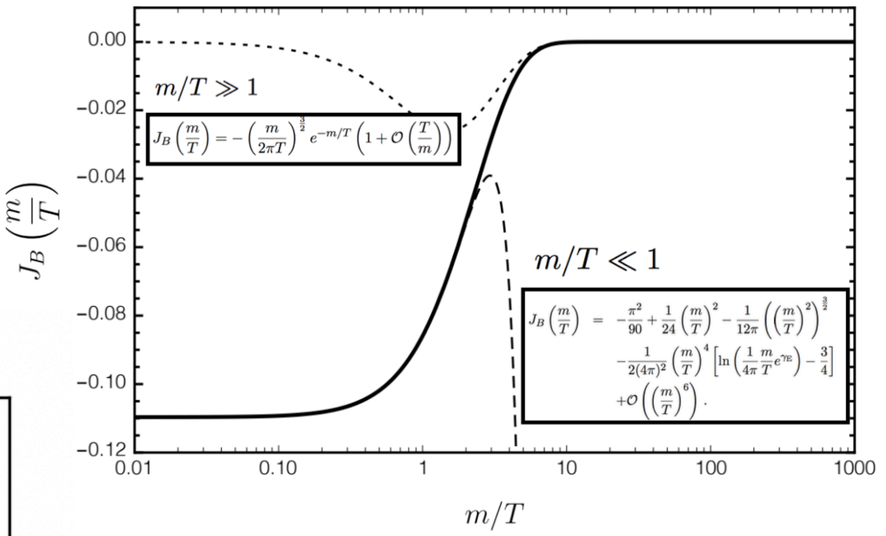


希格斯有效势

- 有限温度下的圈图修正

$$V_{1T}(\phi,T)=\frac{T^4}{2\pi^2}n_{dof}J_B\left(\frac{m^2(\phi)}{T^2}\right),\quad J_B(x)\simeq-\frac{\pi^4}{45}+\frac{\pi^2}{12}x$$





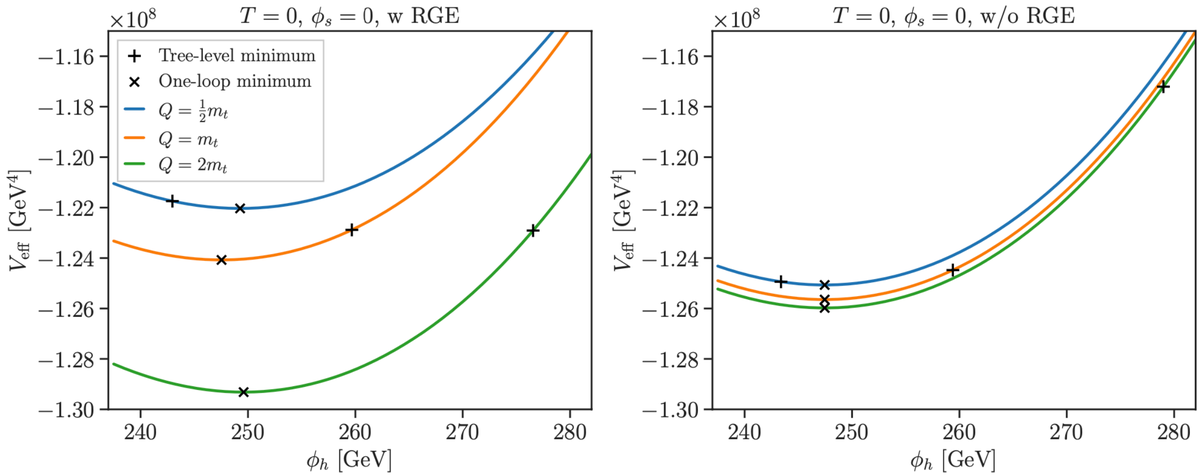
希格斯有效势

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

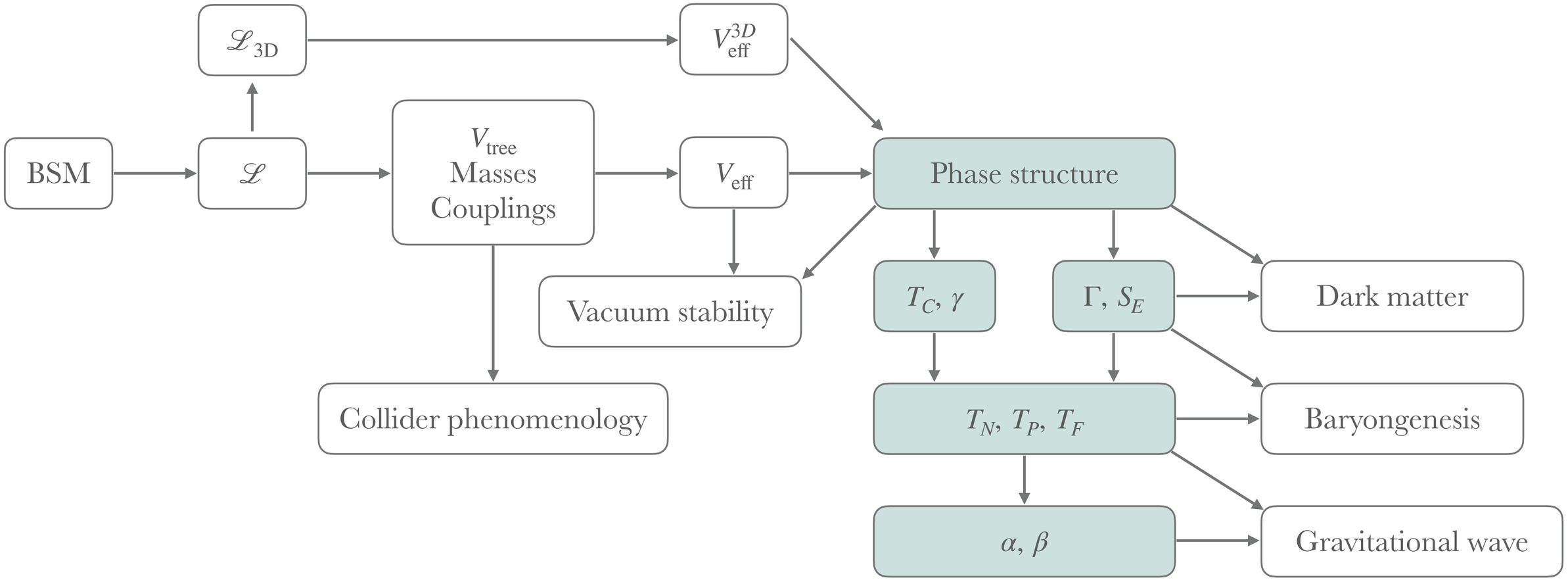
- 有效势中存在的问题：
  - 能标依赖
  - 规范依赖
  - Linde's IR problem
  - Goldstone 灾难
  - Tadpole 限制
- 这些问题很重要，但是不要让他们成为入门的障碍。

arXiv:2208.01319 Peter Athron, Csaba Balazs, Andrew Fowlie, Lachlan Morris, Graham White, Yang Zhang

Name	Order	ξ dependence	μ dependence	Concern
PRM	Tree minima; 1-loop potential	Tadpoles	Explicit	No daisies
High-T	1-loop leading terms	Tadpoles	Implicit	Accuracy
$\overline{\text{MS}}$	1-loop potential	Explicit	Explicit	ξ & μ dependence
$\overline{\text{MS}}$ + RGE	1-loop potential; 2-loop RGE	Explicit	Reduced by RGE	ξ dependence
OS	1-loop potential	Explicit	No	ξ dependence
Covariant gauge	1-loop potential	Explicit	Explicit	ξ & μ dependence

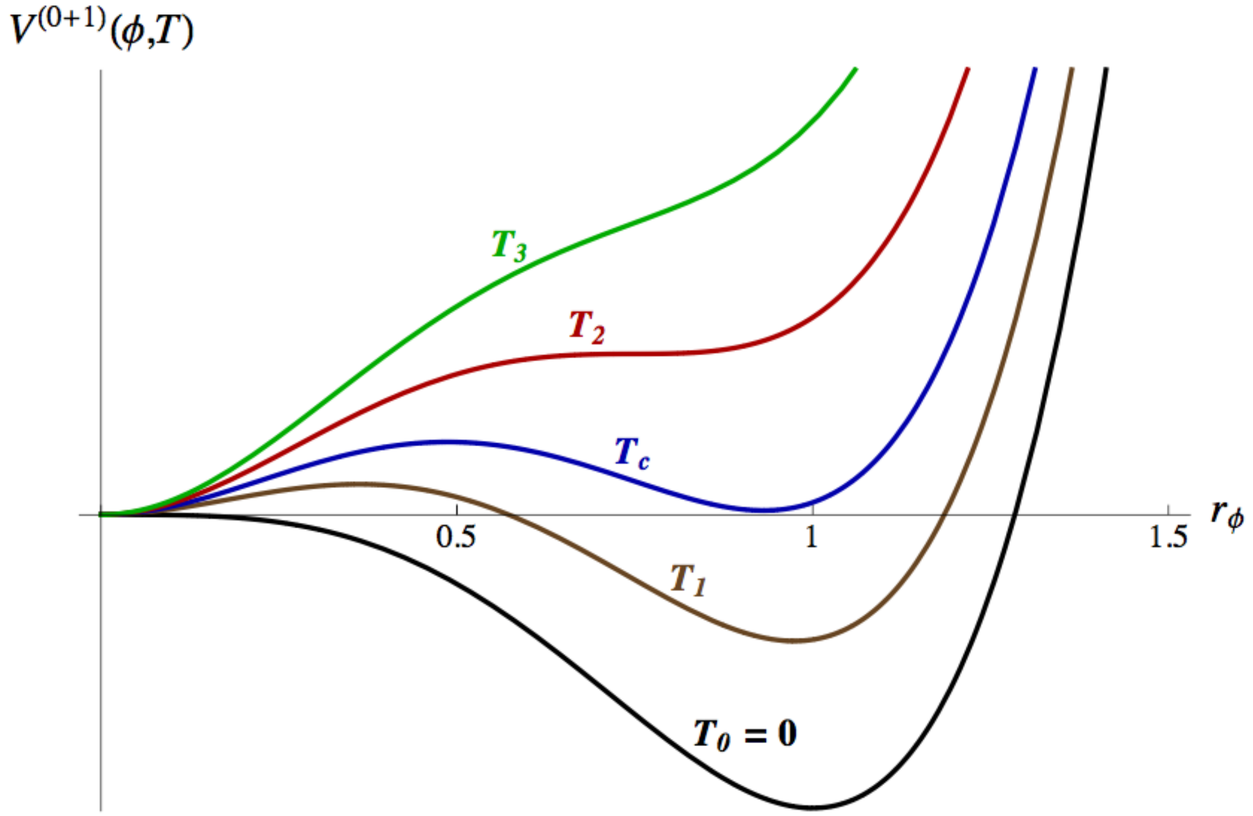


相变参数


$\mathscr{L}_{3D}$
V_{eff}^{3D}
BSM
$\mathscr{L}$
V_{tree}
Masses
Couplings
V_{eff}
Phase structure
Vacuum stability
T_C, γ
Γ, S_E
Dark matter
Collider phenomenology
T_N, T_P, T_F
Baryongenesis
α, β
Gravitational wave

相变参数

- 关键温度 T_C



$$V_{\rm eff}(\phi_i^{(1)}, T_C; \xi) - V_{\rm eff}(\phi_i^{(2)}, T_C; \xi) = 0$$

$$\left.\frac{\partial V_{\rm eff}}{\partial \phi_i}\right|_{\phi_i^{(1)}, T_C} = \left.\frac{\partial V_{\rm eff}}{\partial \phi_i}\right|_{\phi_i^{(2)}, T_C} = 0\,.$$

相变参数

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

➤ 关键温度 T_C

Method 1

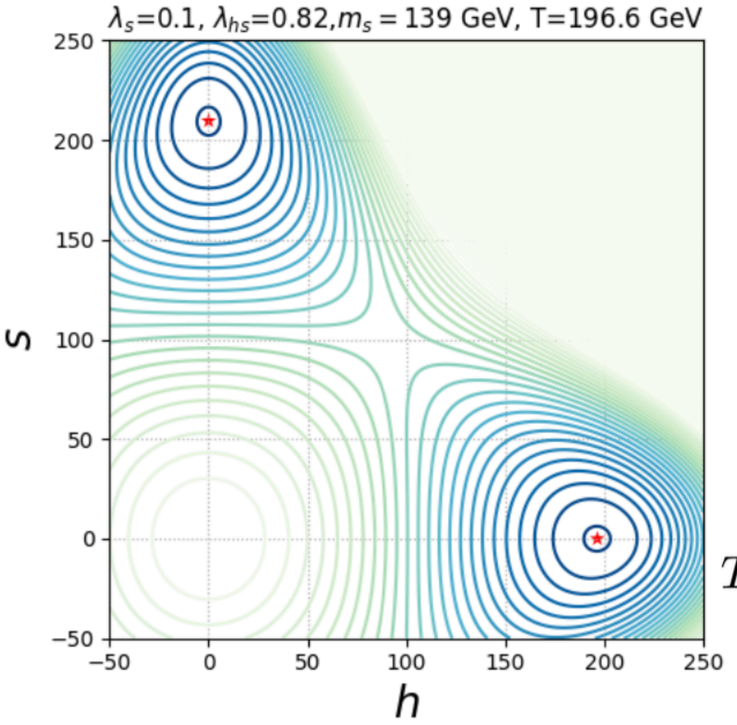
Analytic calculation

Method 2

Method 3

➤ SM + a Z2 symmetric real scalar singlet (arXiv:1611.02073)

$$V(H,s,T)=(\mu_h^2+c_hT^2)H^\dagger H+\lambda_h(H^\dagger H)^2+\frac{\lambda_{hs}}{2}(H^\dagger H)s^2+\frac{(\mu_s^2+c_sT^2)}{2}s^2+\frac{\lambda_s}{4}s^4$$



$$\left.\frac{\partial V}{\partial v}\right|_{v_h,v_s=0}=\left.\frac{\partial V}{\partial v}\right|_{v_s,v_h=0}=0$$

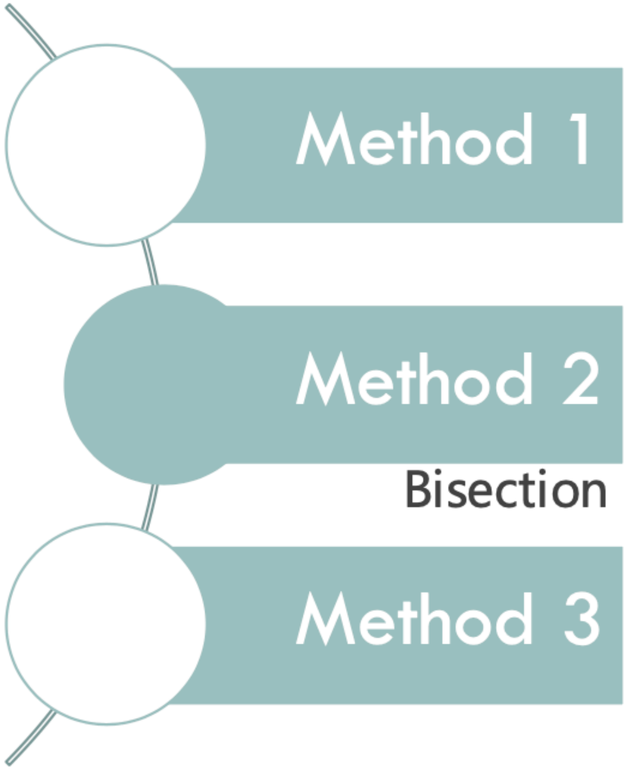
$$v_h=-\frac{\mu_h^2+c_hT^2}{\lambda_h},\ v_s=-\frac{\mu_s^2+c_sT^2}{\lambda_s}$$

$$V(v_h,0,T_c)=V(0,v_s,T_c)$$

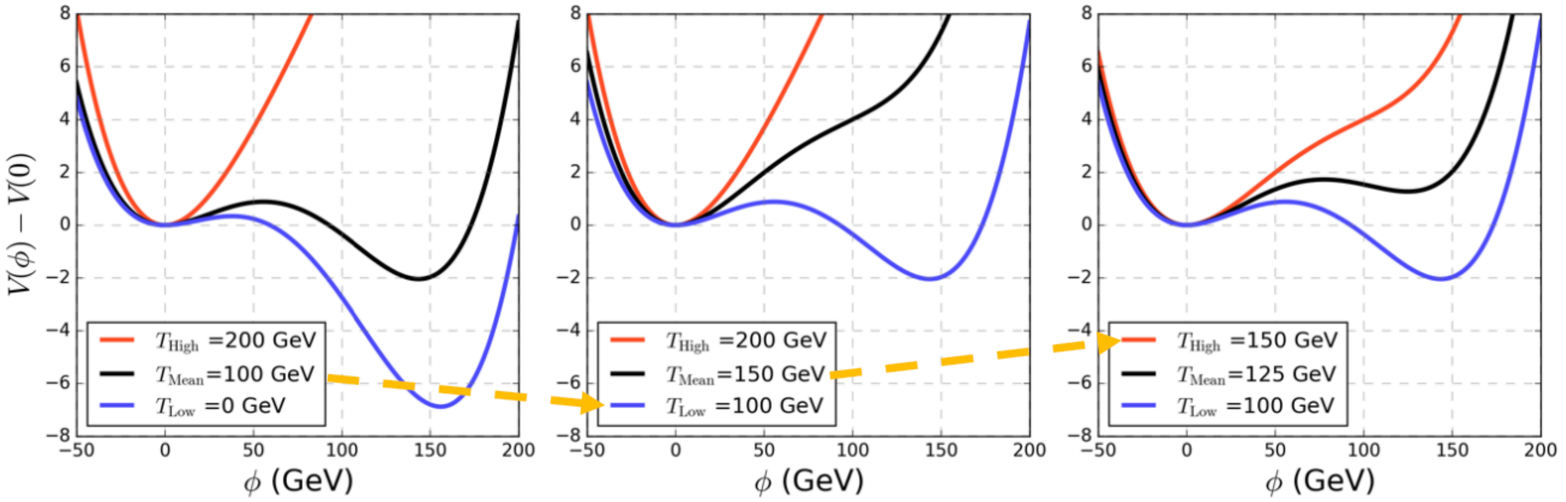
$$T_c^2=\frac{\lambda_sc_h\mu_h^2-\lambda_hc_s\mu_s^2-\sqrt{\lambda_h\lambda_s}|c_s\mu_h^2-c_h\mu_s^2|}{\lambda_sc_h^2-\lambda_hc_s^2}$$

相变参数

➤ 关键温度 T_C



Starting with an un-broken phase at high temperature and a broken phase at low temperature, find a temperature that these two phase degenerate using bisection method.

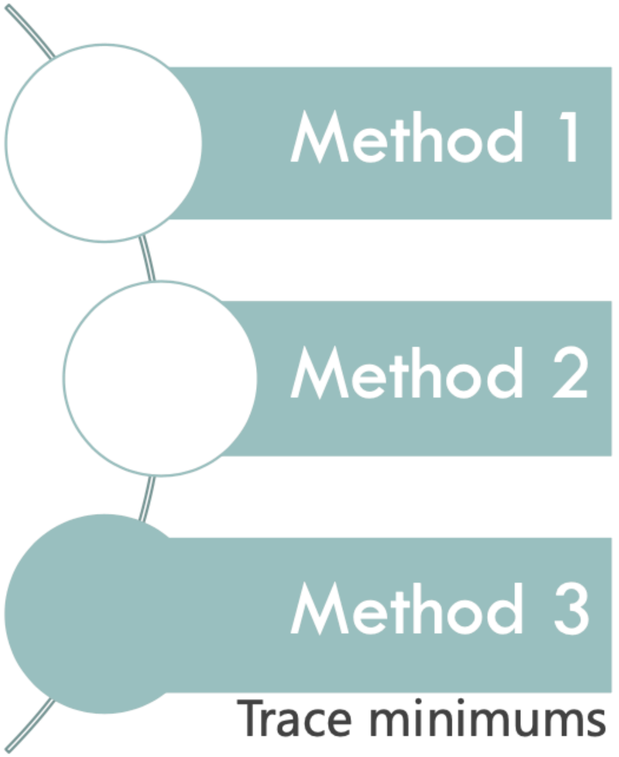


➤ Used in BSMPT (arXiv: 1803.02846)

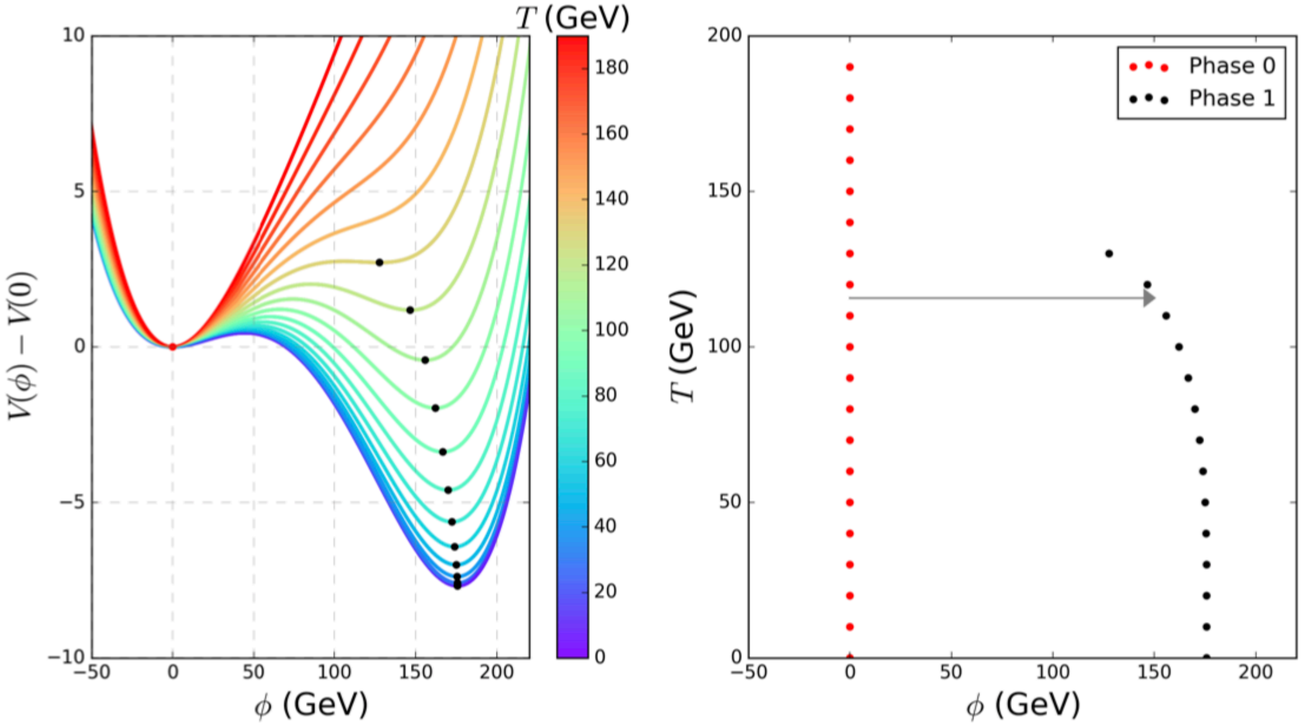
V1，V2

相变参数

- 关键温度 T_C



Starting with a minimum at low/high temperature, trace the location of minimum at different temperatures.



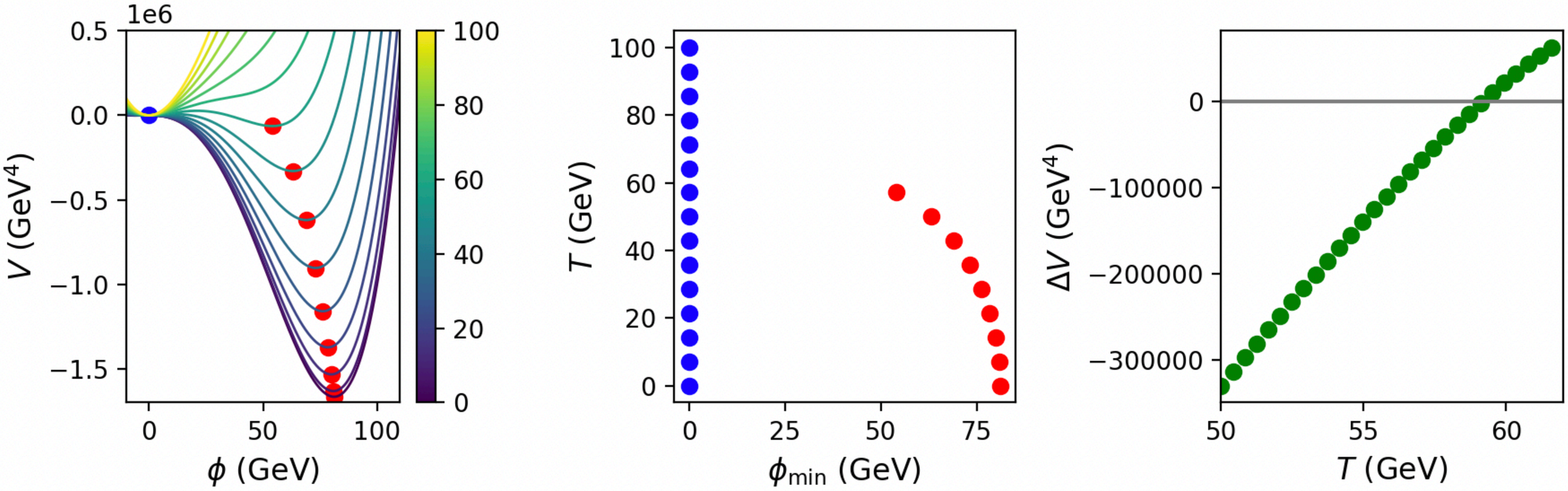
- Used in CosmoTransitions (arXiv: 1109.4189)

PhaseTracer, BSMPT_3

相变参数

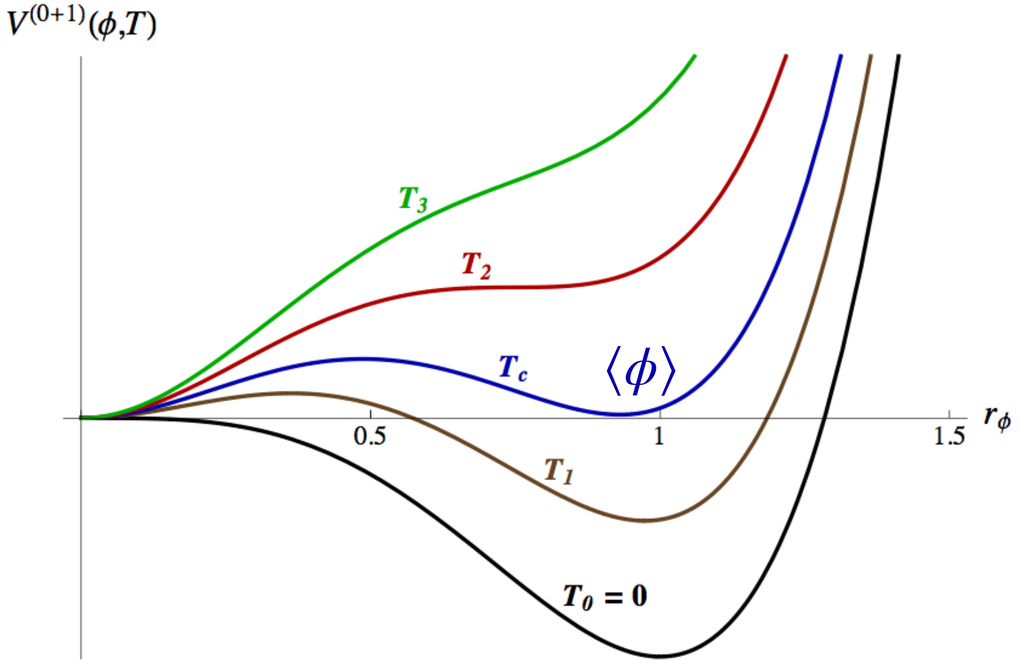
河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

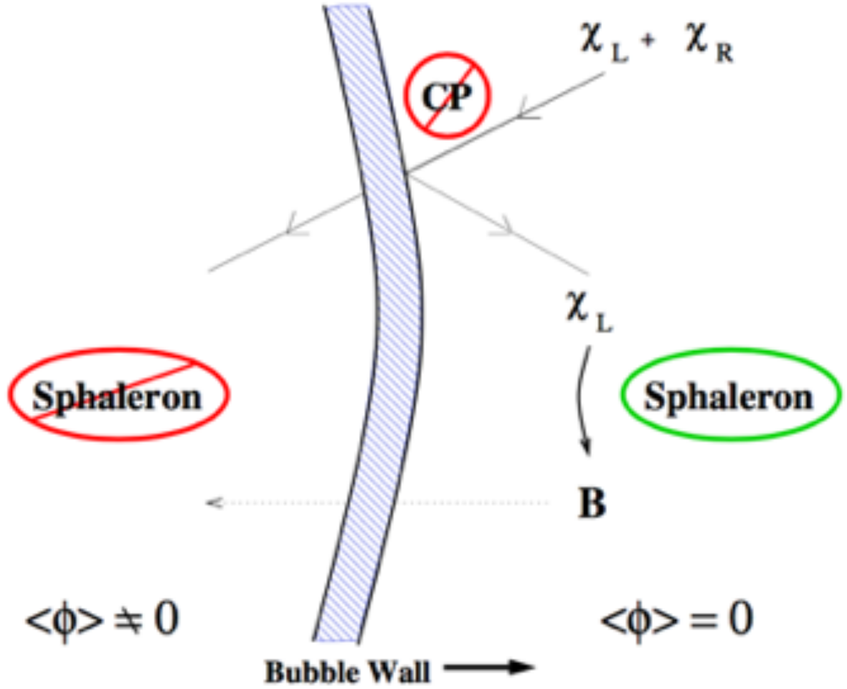
➤ 例：$V(\phi, T) = (cT^2 - m^2)\phi^2 + \kappa\phi^3 + \lambda\phi^4, \quad c = 0.1,\ m^2 = 100,\ \kappa = -10,\ \lambda = 0.1$



相变参数

- 相变强度 $\gamma = \dfrac{\langle\phi\rangle}{T_C}$





$$\frac{E_{\rm sph}(T_c)}{T_c} > 37 \to \gamma \equiv \frac{\phi(T_c)}{T_c} \approx \frac{1}{36}\frac{E_{\rm sph}(T_c)}{T_c} > 1.0$$

相变参数

- 成核率，弹跳做用量 S_E

$$\frac{\Gamma}{V} = A(T)e^{-S_E/T}[1 + \mathcal{O}(\hbar)]$$

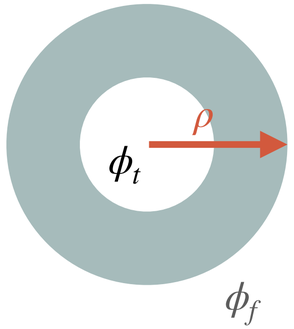
$$S_E = \mathcal{S}_{d-1}\int_0^\infty \rho^{d-1}\left(\frac{1}{2}\dot{\phi}^2 + V(\phi)\right)$$

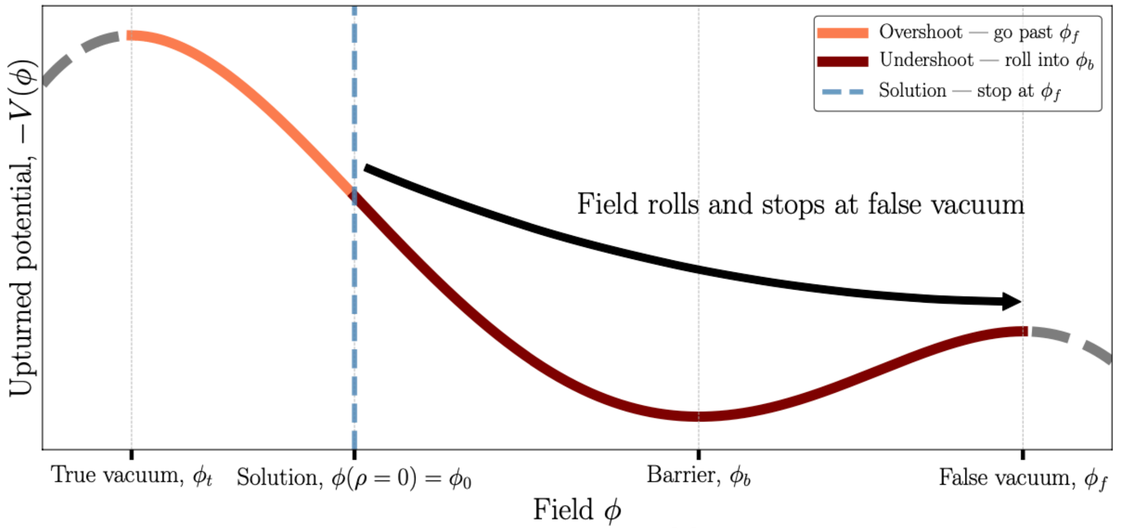
$$\frac{\mathrm{d}^2\phi(\rho)}{\mathrm{d}\rho^2} + \frac{\alpha}{\rho}\frac{\mathrm{d}\phi(\rho)}{\mathrm{d}\rho} = \Delta V(\phi)$$

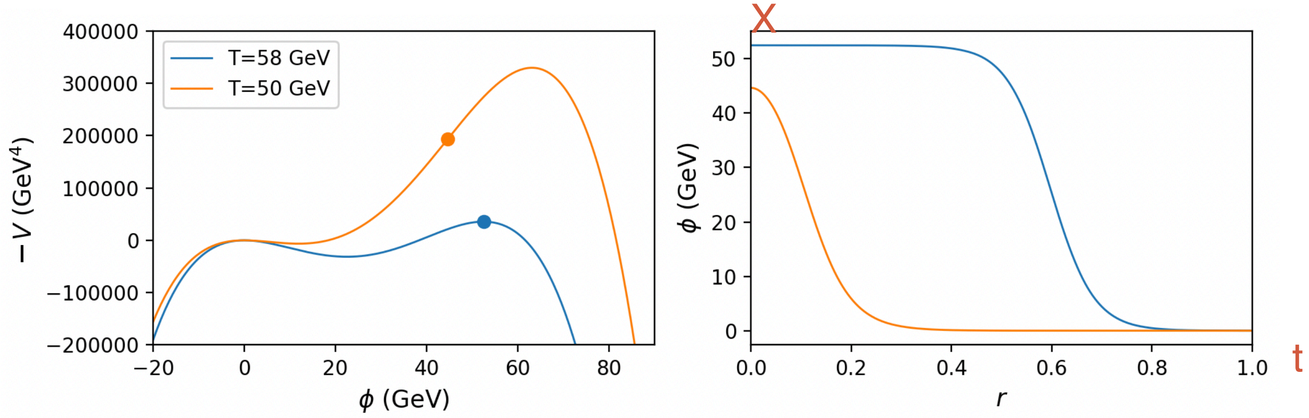
$$\left.\frac{\mathrm{d}\phi(\rho)}{\mathrm{d}\rho}\right|_{\rho=0} = 0$$

$$\phi(\rho \to \infty) = \phi_f$$

$$\left.\frac{\mathrm{d}\phi(\rho)}{\mathrm{d}\rho}\right|_{\rho=\infty} = 0$$





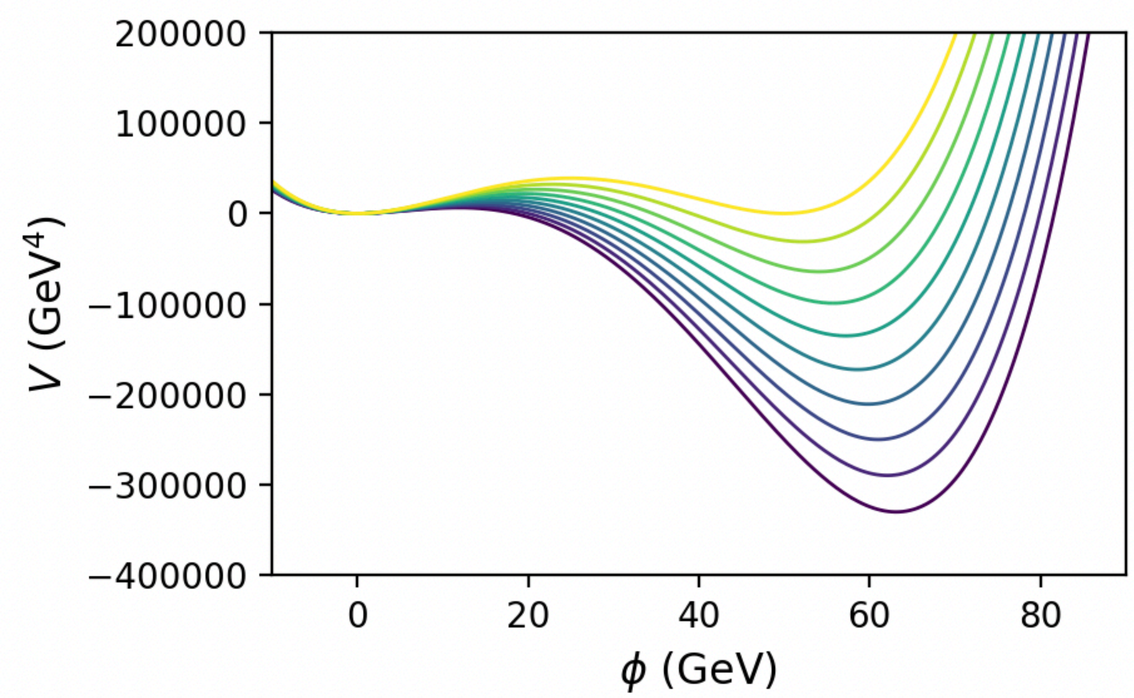


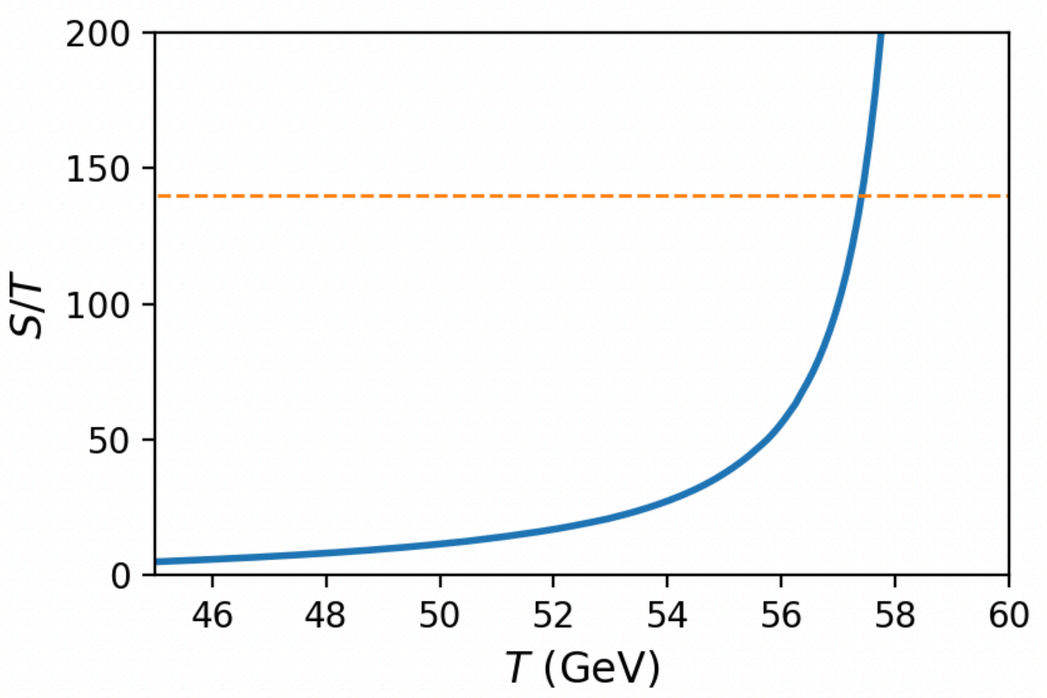
$$\rightarrow \frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{\alpha}{t}\frac{\mathrm{d}x}{\mathrm{d}t} = -\Delta V(x),\ v(t=0) = 0,\ x(t \to \infty) = x_f,\ v(t \to \infty) = 0$$

相变参数

- 成核温度 T_N

$$N(T) = \int_{T_C}^{T} dT' \frac{\Gamma(T')P_f(T')}{T'H^4(T')} = 1 \qquad \frac{S_E(T_N)}{T_N} \sim 140$$



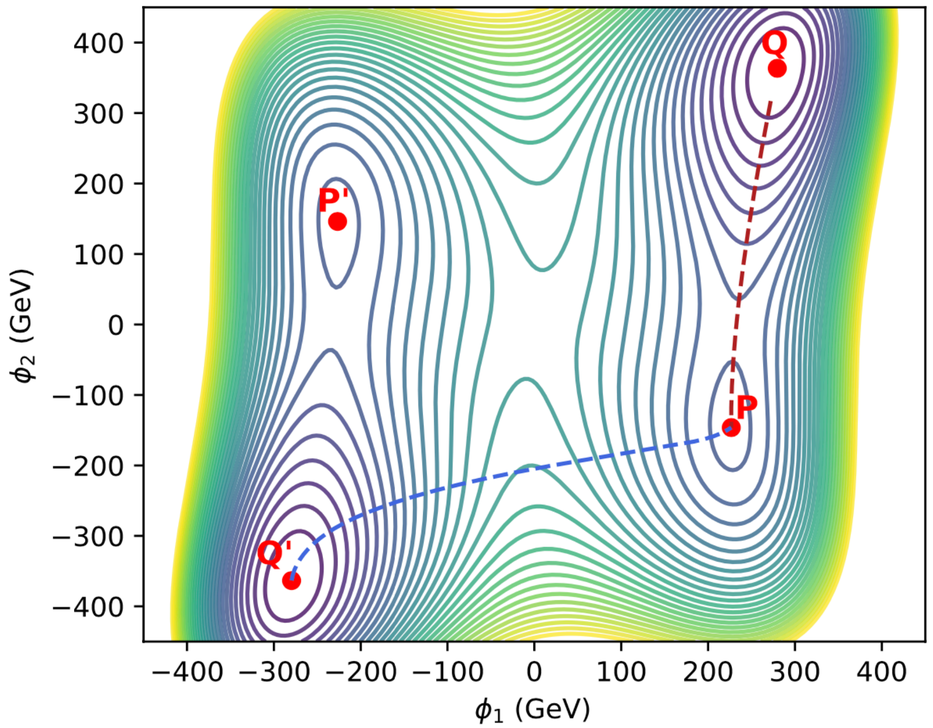


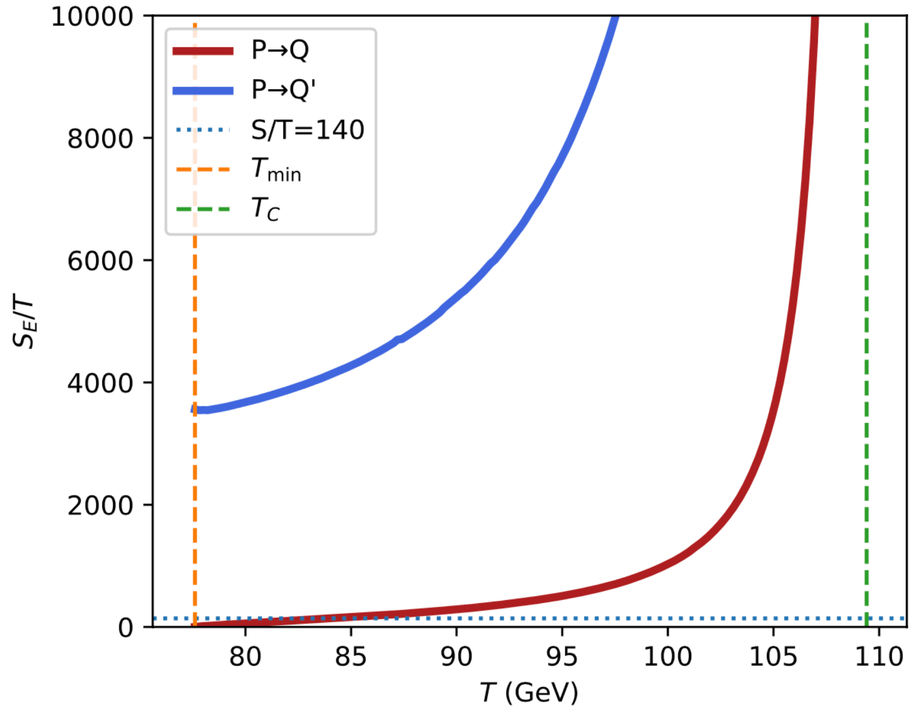
相变参数

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

- 分离对称性

$$V(\phi_1, \phi_2) = V(-\phi_1, -\phi_2)$$





- In CosmoTransitions, half of the space is forbidden.
- In BSMPT3, the one with shorter distance is chosen.
- In PhaseTracer2, the one with minimum action is chosen.

相变参数

- 潜热/相变强度：$$\alpha = \frac{D\theta}{\pi^2 g_* T_*^4/30} = \frac{1}{\pi^2 g_* T_*^4/30}\left(V(\phi) - \frac{T}{4}\frac{\partial V(\phi,T)}{\partial T}\right)\Bigg|_{\phi_t}^{\phi_f}$$

- 相变弛豫时间：$$\beta(T) = \frac{d}{dt}\left(\frac{S_E}{T}\right) = TH(T)\frac{d}{dT}\left(\frac{S_E}{T}\right)$$

- 相变引力波谱：
  - Bubble collisions

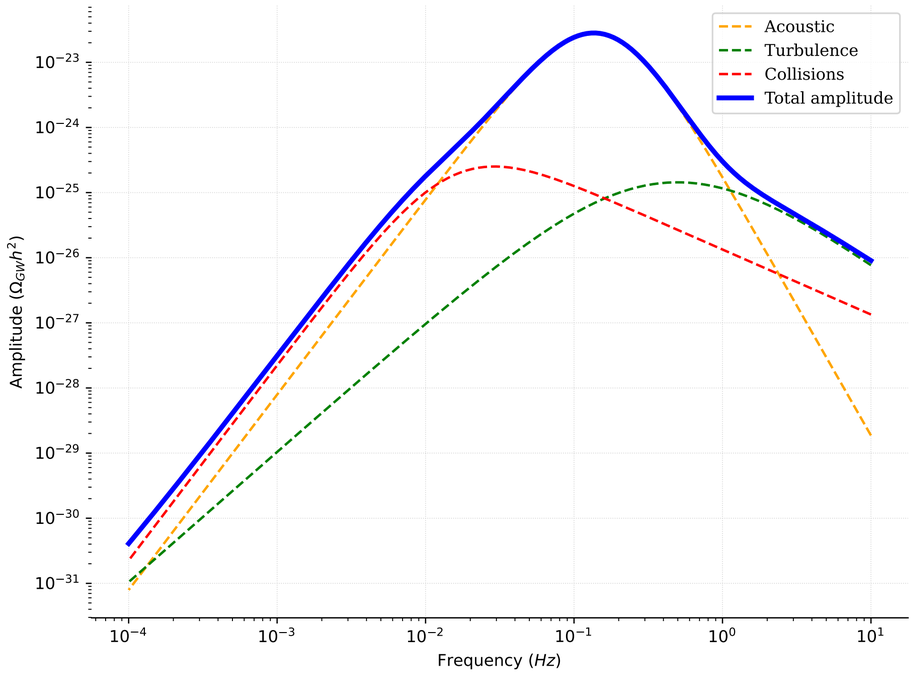
$$\Omega_{\rm col}^{\rm env} h^2 = 1.67\times 10^{-5}\Delta\left(\frac{100}{g_*}\right)^{\frac{1}{3}}\left(\frac{H_*}{\beta}\right)^2\left(\frac{\kappa_\phi \alpha}{1+\alpha}\right)^2 \times S_{\rm env}(f), \quad (82)$$

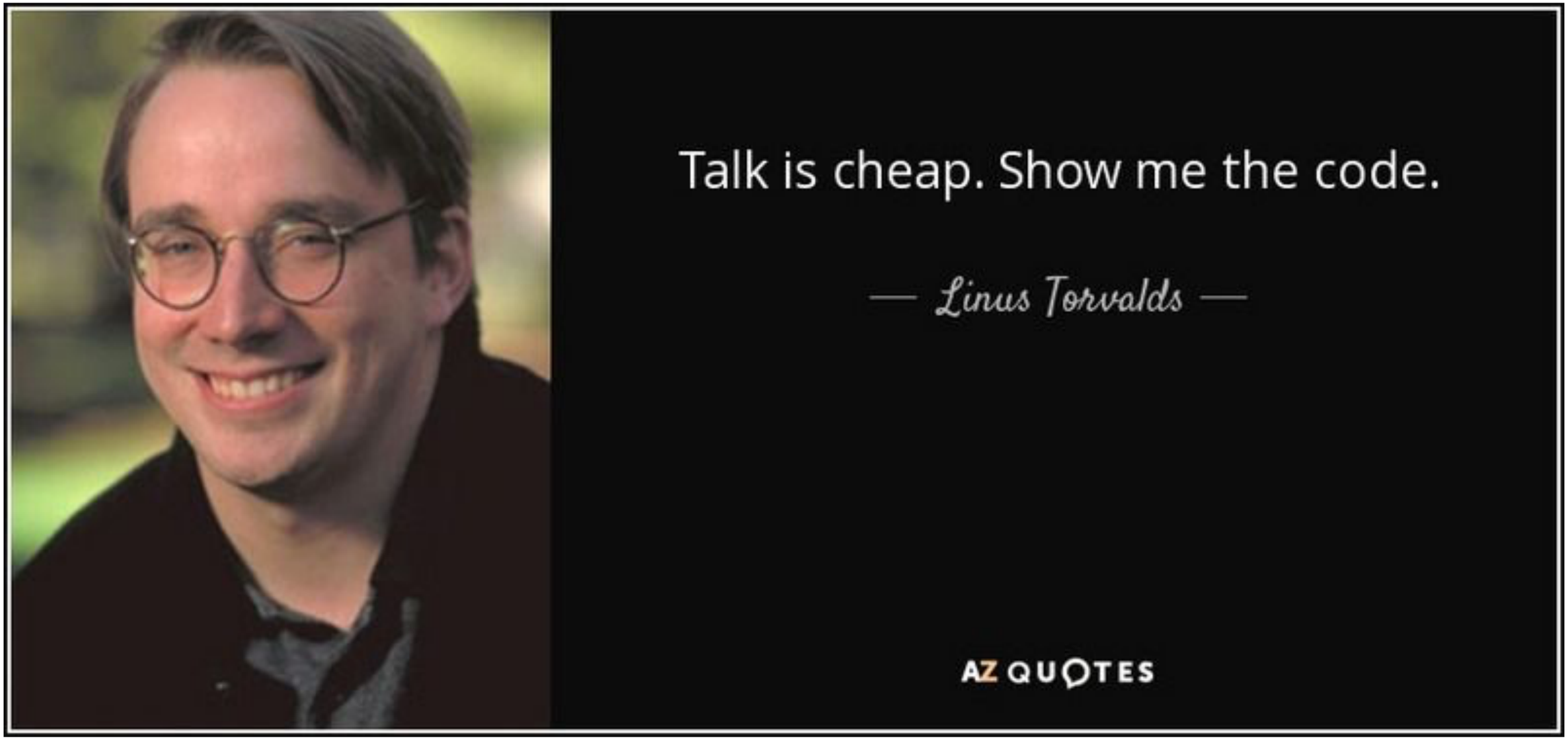
  - Turbulence

$$\Omega_{\rm turb}h^2 = 3.35\times 10^{-4}\left(\frac{H_*}{\beta}\right)\left(\frac{\kappa_{\rm turb}\alpha}{1+\alpha}\right)^{3/2}\left(\frac{100}{g_*}\right)^{1/3} v_w \times \frac{(f/f_{\rm turb})^3}{[1+(f/f_{\rm turb})]^{11/3}(1+8\pi f/H_0)}, \quad (88)$$

  - Sound waves

$$\Omega_{\rm sw}h^2 = 2.061\, F_{\rm gw,0}\,\Gamma^2\,\bar{U}_f^4\, S_{\rm sw}(f)\,\tilde{\Omega}_{\rm gw} \times \min(H_*R_*/\bar{U}_f, 1)\,(H_*R_*)\,h^2, \quad (91)$$




Talk is cheap. Show me the code.
— Linus Torvalds —
AZ QUOTES

宇宙学相变计算工具

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

- CosmoTransitions：Python，traces minima，calculate the bounce solution，nucleation temperature
- BSMPT_v1: calculate critical temperature
- PhaseTracer_v1: tracks phases, calculate critical temperature
- BSMPT_v2: baryon asymmetry
- BSMPT_v3: tracks phases, calculate the bounce solution, nucleation/percolation/completion temperature, GW
- PhaseTracer_v2: calculate the bounce solution, nucleation/percolation/completion temperature, GW

- Vevacious：Python，find all extrema at T=0, calculate tunnelling times using CosmoTransitions
- VevaciousPlusPlus: C++ code wrapper of CosmoTransitions
- EVADE: estimates the decay rate of the false vacuum

- AnyBubble：Mathematica, calculate the bounce solution
- BubbleProfiler: C++, calculate the bounce solution
- SimpleBounce: calculate the bounce solution
- FindBounce: calculate the bounce solution
- OptiBounce: calculate the bounce solution

PhaseTracer2

FlexibleSUSY

DRalgo

Input: Effective potential

- $\overline{\text{MS}}$ Scheme
- HT Expansion
- OS-like Scheme
- 3D EFT
- R_ξ Gauge
- Covariant Gauge
- Arnold-Espinosa Approach
- Parwani Approach

BubbleProfiler

TransitionSolver

Outputs

- Phase structure
- Critical Temperature
- Bounce Action
- Nucleation Temperature
- Strength factor
- Transition duration
- Gravitational Wave Spectrum
- Gravitational Wave SNR

Inputs of PhaseTracer2

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

- Model: SM, Singlet extension, 2HDM, NMSSM
- Input: V_{tree}, num of fields, symmetry, field-dependent masses, Debye masses

```
#include "one_loop_potential.hpp"
#include "pow.hpp"
#include "SM_parameters.hpp"


namespace EffectivePotential {

class xSM_base : public OneLoopPotential {
 public:
  double V0(Eigen::VectorXd phi) const override {
    return 0.5 * muh_sq * square(phi[0]) +
           0.25 * lambda_h * pow_4(phi[0]) +
           0.25 * lambda_hs * square(phi[0]) * square(phi[1]) +
           0.5 * mus_sq * square(phi[1]) +
           0.25 * lambda_s * pow_4(phi[1]);
  }
  size_t get_n_scalars() const override {
    return 2;
  }
  std::vector<Eigen::VectorXd> apply_symmetry(Eigen::VectorXd phi) const override {
    auto phi1 = phi;
    phi1[0] = - phi[0];
    auto phi2 = phi;
    phi2[1] = - phi[1];
    return {phi1, phi2};
  };
```

$$V_0(h,s) = \frac{\mu_H^2}{2}h^2 + \frac{\lambda_H}{4}h^4 + \frac{\mu_S^2}{2}s^2 + \frac{\lambda_S}{4}s^4 + \frac{\lambda_{HS}}{4}h^2s^2$$

$$n_{\text{scalars}} = 2$$

$$V(h,s) = V(-h,s), V(h,s) = V(h,-s)$$

Inputs of PhaseTracer2

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

- Model: SM, Singlet extension, 2HDM, NMSSM
- Input: V_{tree}, num of fields, symmetry, field-dependent masses, Debye masses

```
58  std::vector<double> get_fermion_masses_sq(Eigen::VectorXd phi) const override {
59    return {0.5 * SM_yt_sq * square(phi[0]),
60            0.5 * SM_yb_sq * square(phi[0]),
61            0.5 * SM_ytau_sq * square(phi[0])};
62  }
63  std::vector<double> get_fermion_dofs() const override {
64    return {12., 12., 4.};
65  }
66  std::vector<double> get_scalar_thermal_sq(double T) const override {
67    const double c_h = (9. * square(SM_g) +
68                        3. * square(SM_gp) +
69                        2. * (6. * SM_yt_sq + 6. * SM_yb_sq +
70                              2. * SM_ytau_sq + 12. * lambda_h + lambda_hs)) / 48.;
71    const double c_s = (2. * lambda_hs + 3. * lambda_s) / 12.;
72    return {c_h * square(T), c_s * square(T)};
73  }
74  std::vector<double> get_scalar_debye_sq(Eigen::VectorXd phi, double xi, double T) const override{
75    const double h = phi[0];
76    const double s = phi[1];
77    const auto thermal_sq = get_scalar_thermal_sq(T);
78
79    const double mhh2 = muh_sq + 3. * lambda_h * square(h) + 0.5 * lambda_hs * square(s);
80    ... ...
81  }
82  std::vector<double> get_scalar_dofs() const override {
83    return {1., 1., 1., 1., 1};
84  }
```

$$m^2_{b,t,\tau}(\phi) = \frac{y^2_{b,t,\tau}}{2}\phi_h^2$$

$$n_t = 12,\ n_b = 12,\ n_\tau = 4$$

$$\overline{m}_h^2(\phi,T) = -\left(\mu_h^2+\mu_s^2\right) + 3\left(\lambda_h\phi_h^2+\lambda_s\phi_s^2\right) + \frac{1}{2}\lambda_{hs}\left(\phi_h^2+\phi_s^2\right) + \left(c_h+c_s\right)T^2 + \left[\left(-\left(\mu_h^2-\mu_s^2\right) + 3\left(\lambda_h\phi_h^2-\lambda_s\phi_s^2\right) - \frac{1}{2}\lambda_{hs}\left(\phi_h^2-\phi_s^2\right) + \left(c_h-c_s\right)T^2\right)^2 + 4\lambda_{hs}^2\phi_h^2\phi_s^2\right]^{\frac{1}{2}}, \quad (50)$$

$$\overline{m}_s^2(\phi,T) = -\left(\mu_h^2+\mu_s^2\right) + 3\left(\lambda_h\phi_h^2+\lambda_s\phi_s^2\right) + \frac{1}{2}\lambda_{hs}\left(\phi_h^2+\phi_s^2\right) + \left(c_h+c_s\right)T^2 - \left[\left(-\left(\mu_h^2-\mu_s^2\right) + 3\left(\lambda_h\phi_h^2-\lambda_s\phi_s^2\right) - \frac{1}{2}\lambda_{hs}\left(\phi_h^2-\phi_s^2\right) + \left(c_h-c_s\right)T^2\right)^2 + 4\lambda_{hs}^2\phi_h^2\phi_s^2\right]^{\frac{1}{2}}, \quad (51)$$

$$m_G^2(\phi,T) = \mu_h^2 + \lambda_h\phi_h^2 + \frac{1}{2}\lambda_{hs}\phi_s^2 + c_hT^2, \quad (52)$$

$$n_h = 1,\ n_s = 1,\ n_G = 3$$

Inputs of PhaseTracer2

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

- Model: SM, Singlet extension, 2HDM, NMSSM
- Input: V_{tree}, num of fields, symmetry, field-dependent masses, Debye masses

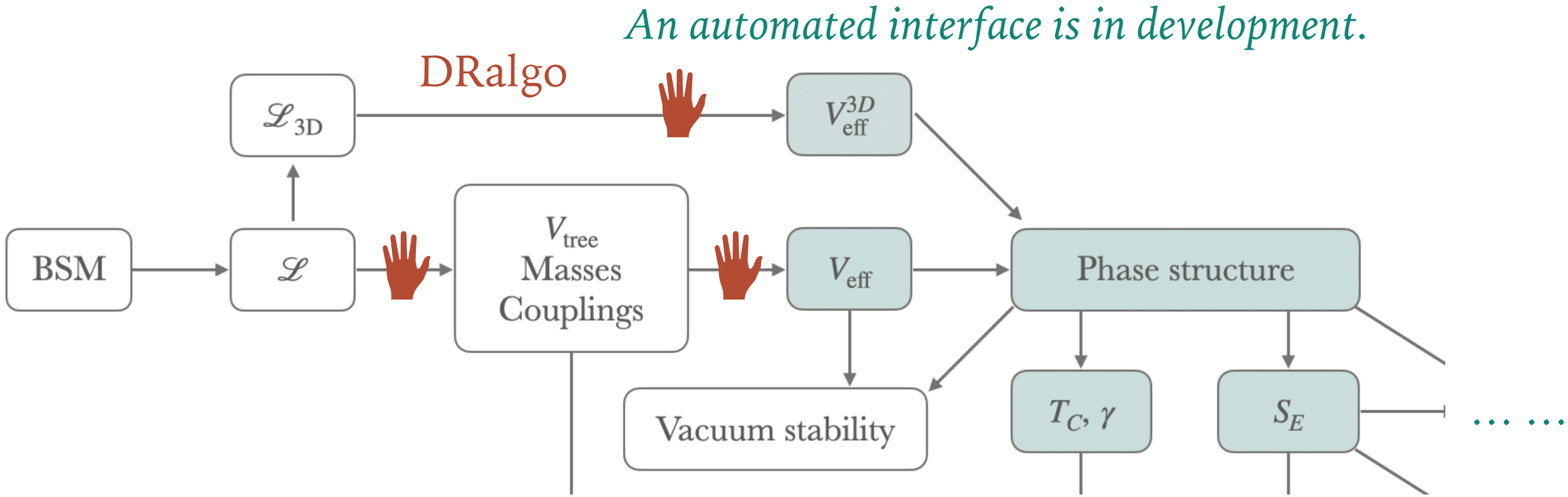
```
91   // W, Z, photon
92   std::vector<double> get_vector_debye_sq(Eigen::VectorXd phi, double T) const override {
93     const double h_sq = square(phi[0]);
94     const double T_sq = square(T);
95     const double MW_T_sq = 0.25 * square(SM_g) * h_sq;
96     const double MZ_T_sq = 0.25 * (square(SM_g) + square(SM_gp)) * h_sq;
97     const double Mphoton_T_sq = 0.;
98
99     const double MW_L_sq = 0.25 * square(SM_g) * h_sq + 11. / 6. * square(SM_g) * T_sq;
100    const double a_L = (square(SM_g) + square(SM_gp)) * (3. * h_sq + 22. * T_sq);
101    const double b_L = std::sqrt(9. * square(square(SM_g) + square(SM_gp)) * square(h_sq)
102                    + 132. * square(square(SM_g) - square(SM_gp)) * h_sq * T_sq
103                    + 484. * square(square(SM_g) - square(SM_gp)) * pow_4(T));
104    const double MZ_L_sq = (a_L + b_L) / 24.;
105    const double Mphoton_L_sq = (a_L - b_L) / 24.;
106
107    // Mphoton_sq must be put at the end, as it will not be used in the OSlike scheme.
108    return {MW_L_sq, MZ_L_sq, Mphoton_L_sq, MW_T_sq, MZ_T_sq, Mphoton_T_sq};
109  }
110
111  // W, Z, photon
112  std::vector<double> get_vector_dofs() const override {
113    return {2., 1., 1., 4., 2, 2};
114  }
```

$$
\begin{aligned}
m^2_{W_T}(\boldsymbol{\phi}) &= \frac{g^2}{4}\phi_h^2, \\
m^2_{W_L}(\boldsymbol{\phi}, T) &= \frac{g^2}{4}\phi_h^2 + \frac{11}{6}g^2T^2, \\
m^2_{\gamma_T}(\boldsymbol{\phi}) &= 0, \\
m^2_{\gamma_L}(\boldsymbol{\phi}, T) &= \frac{1}{24}\Big[\left(g^2+g'^2\right)\left(3\phi_h^2+22T^2\right) \\
&\qquad - \sqrt{9\left(g^2+g'^2\right)^2\phi_h^4 + 44T^2\left(g^2-g'^2\right)^2\left(3\phi_h^2+11T^2\right)}\Big], \\
m^2_{Z_T}(\boldsymbol{\phi}) &= \frac{g^2+g'^2}{4}\phi_h^2, \\
m^2_{Z_L}(\boldsymbol{\phi}, T) &= \frac{1}{24}\Big[\left(g^2+g'^2\right)\left(3\phi_h^2+22T^2\right) \\
&\qquad + \sqrt{9\left(g^2+g'^2\right)^2\phi_h^4 + 44T^2\left(g^2-g'^2\right)^2\left(3\phi_h^2+11T^2\right)}\Big].
\end{aligned}
$$

$$
n_{W_L} = 2,\ n_{Z_L} = 1,\ n_{\gamma_L} = 1, n_{W_T} = 4,\ n_{Z_T} = 2,\ n_{\gamma_T} = 2
$$

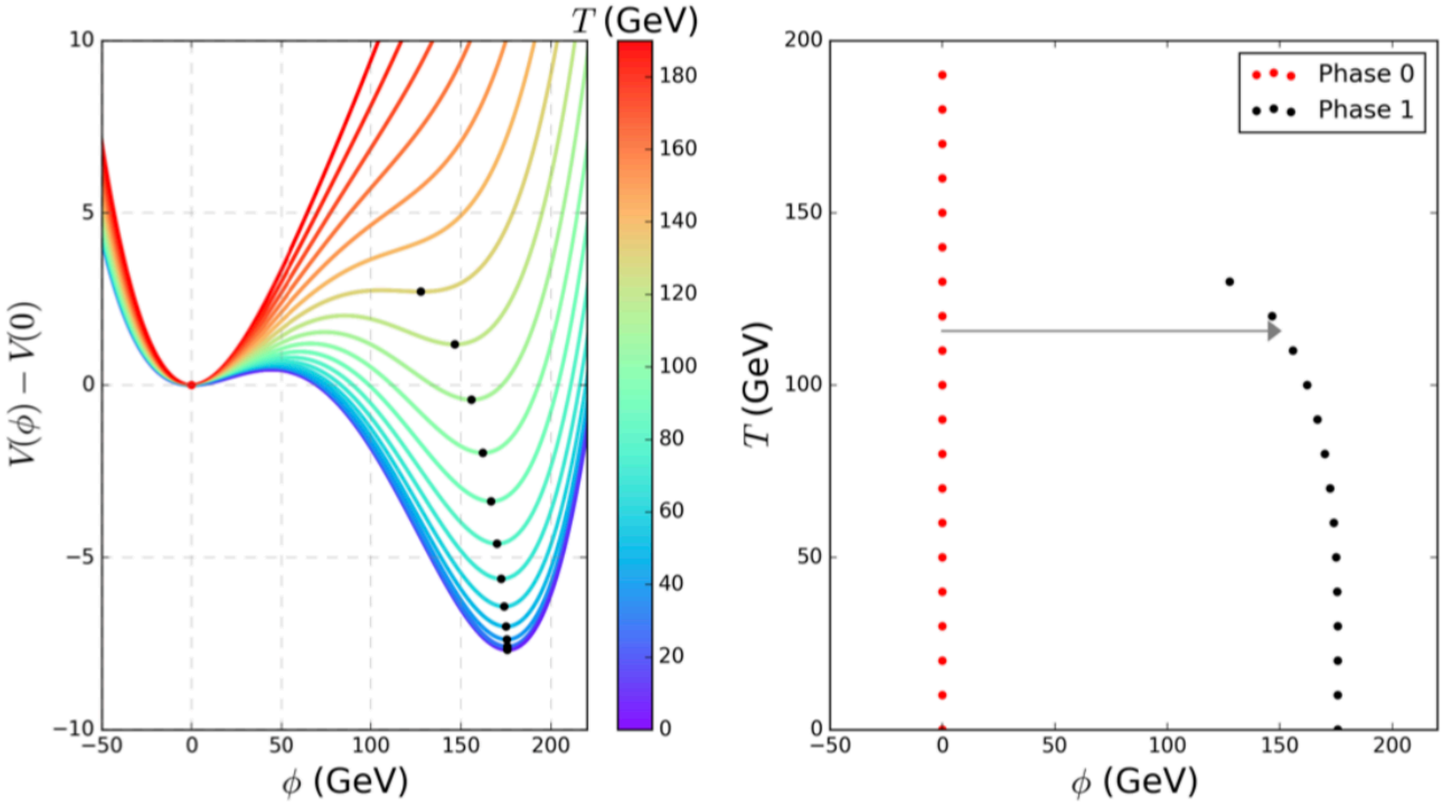
Inputs of PhaseTracer2

- Model: SM, Singlet extension, 2HDM, NMSSM
- Input: V_{eff}^{3D} from DRalgo



Outputs of PhaseTracer2

- Phase structure, transition parameters

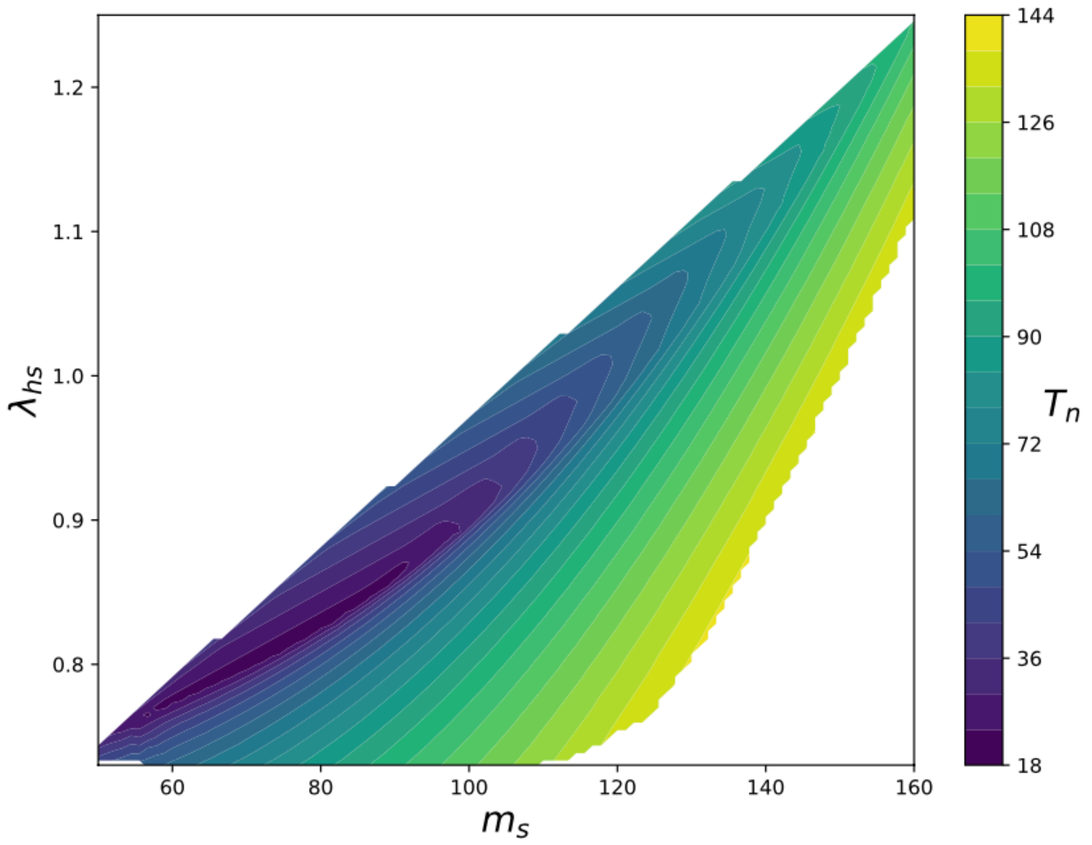


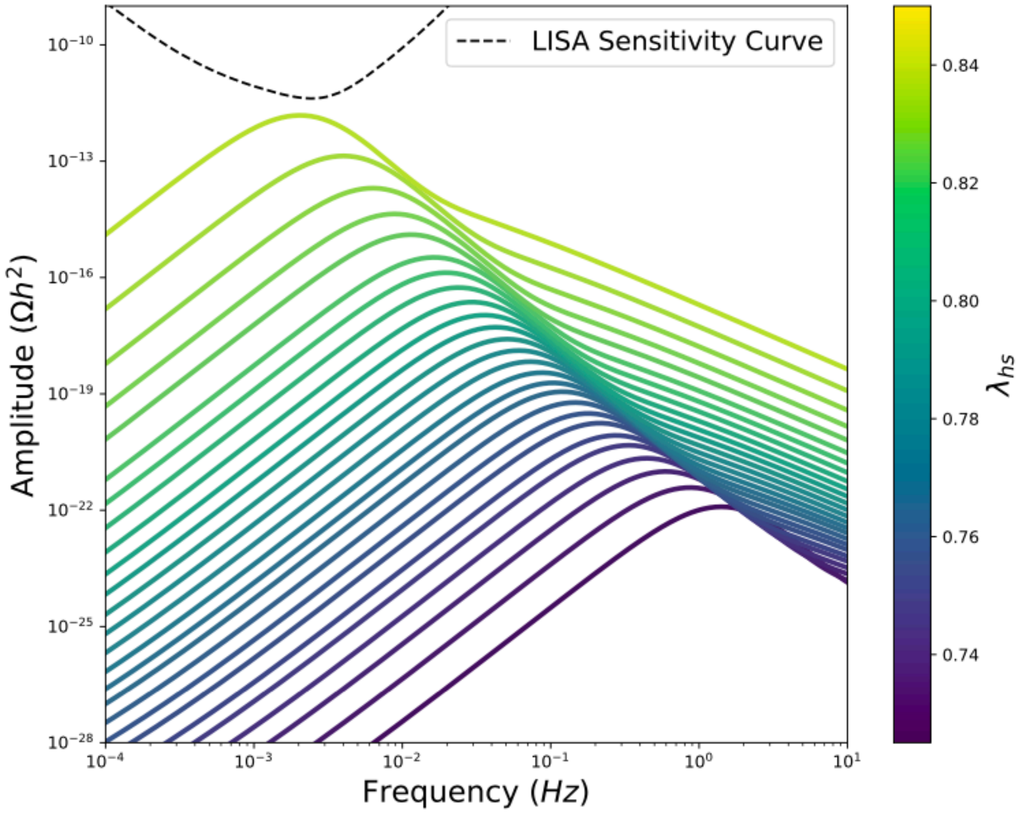

```
PhaseTracer — -zsh — 58×43
zy@192 PhaseTracer % ./bin/run_1D_test_model
found 2 phases

=== phase key = 0 ===
Maximum temperature = 1000
Minimum temperature = 33.1513
Field at tmax = [-7.71149e-06]
Field at tmin = [-4.80989e-06]
Potential at tmax = 5.94077e-06
Potential at tmin = 2.29061e-10
Ended at tmax = Reached tstop
Ended at tmin = Jump in fields indicated end of phase

=== phase key = 1 ===
Maximum temperature = 61.7437
Minimum temperature = 0
Field at tmax = [37.8319]
Field at tmin = [81.1597]
Potential at tmax = 65886.6
Potential at tmin = -1.66587e+06
Ended at tmax = Jump in fields indicated end of phase
Ended at tmin = Reached tstop

found 1 transition

=== transition from phase 0 to phase 1 ===
changed = [true]
TC = 59.1608
false vacuum (TC) = [5.69488e-06]
true vacuum (TC) = [50.0002]
gamma (TC) = 0.845158
delta potential (TC) = 0.00117802
TN = 57.3828
false vacuum (TN) = [6.40552e-06]
true vacuum (TN) = [53.6195]

=== Gravitational wave generated at T = 57.3828
    alpha = 0.00138966
    beta over H = 7118.31
    peak_frequency = 0.121947
    peak_amplitude = 3.59588e-23
    siganl to noise ratio for LISA = 3.72509e-13
zy@192 PhaseTracer %
```

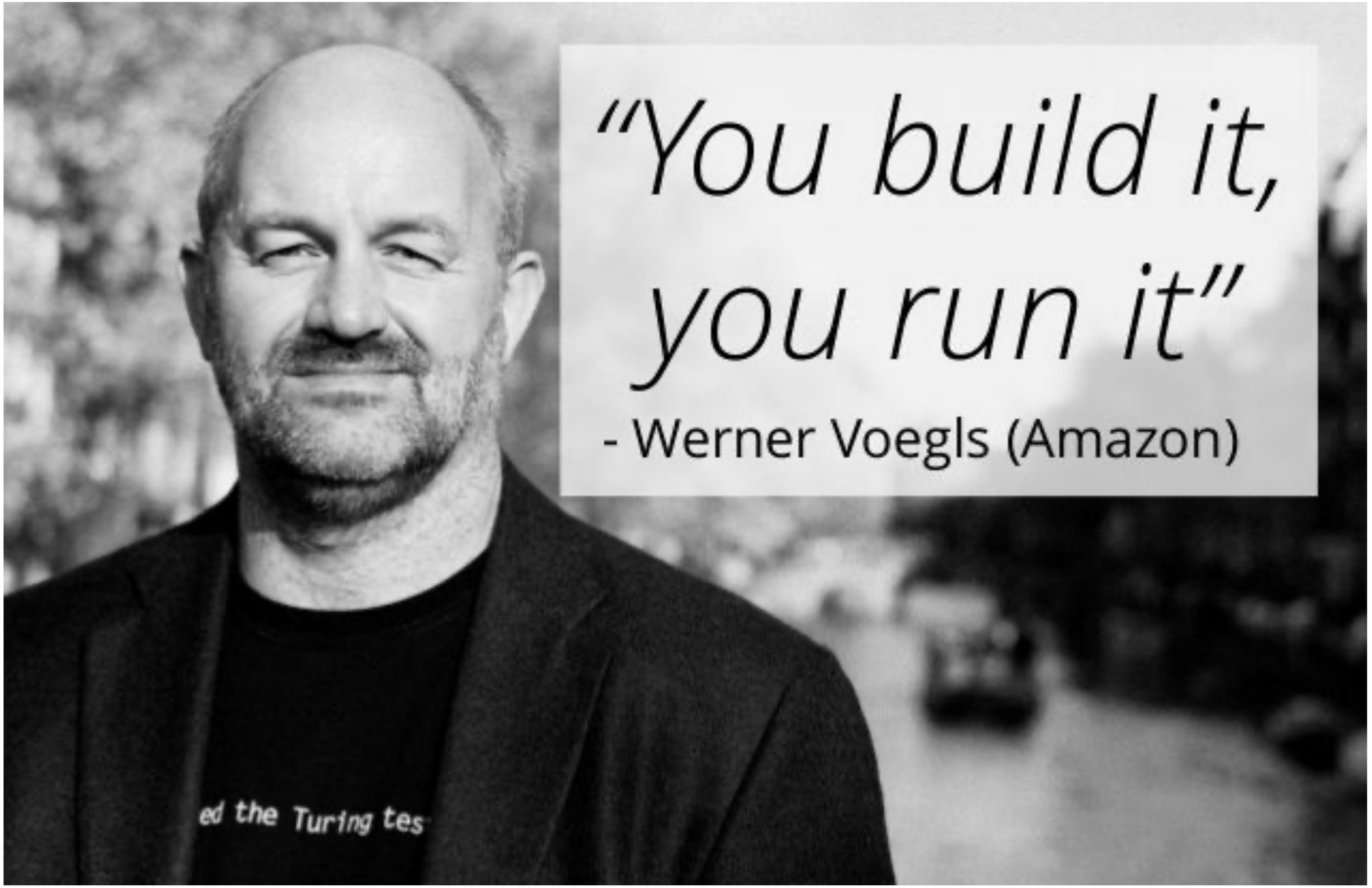
Outputs of PhaseTracer2

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

- GW spectrum

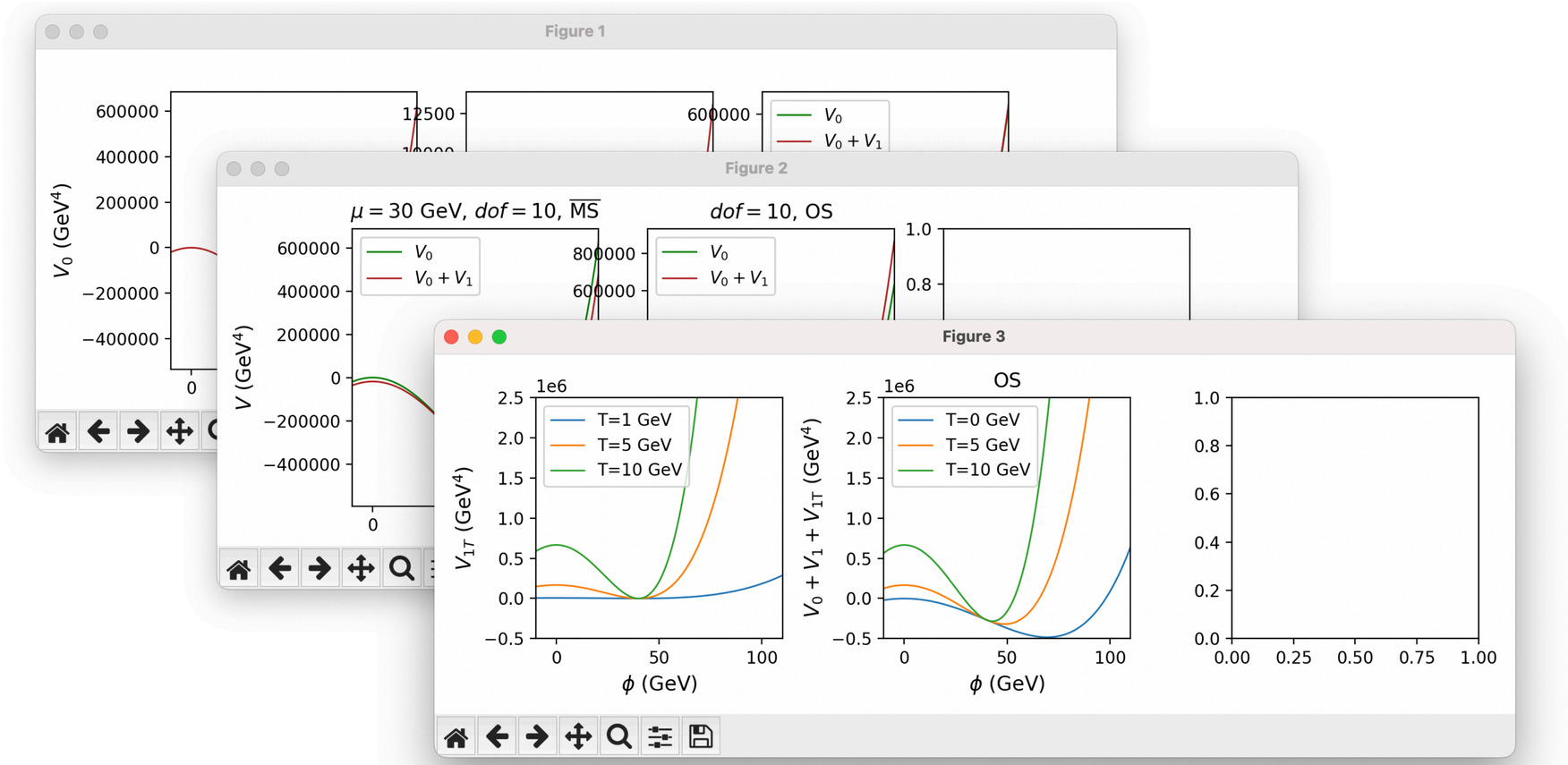





“You build it, you run it”
- Werner Voegls (Amazon)
ed the Turing tes

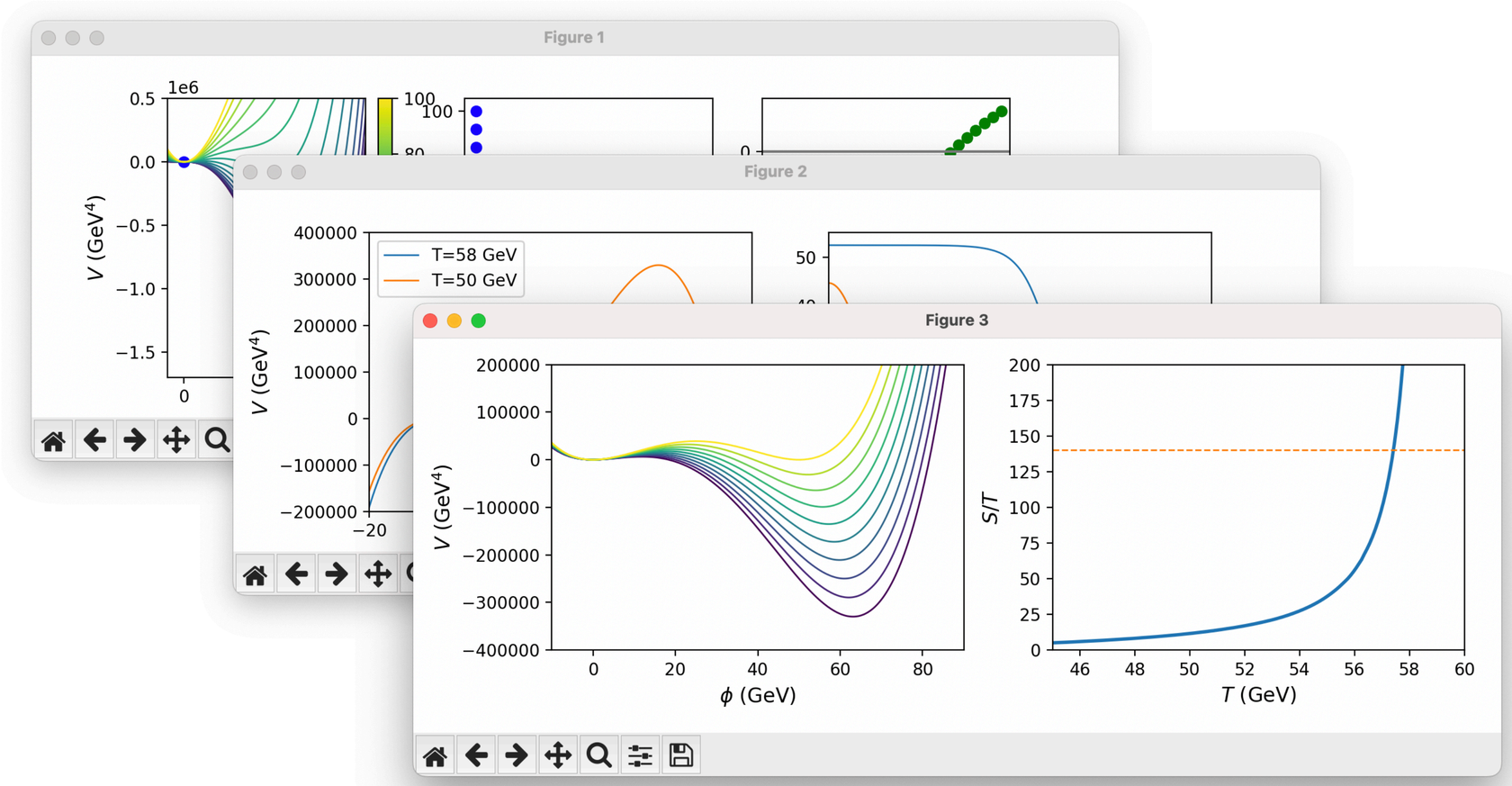
练习 1：ϕ^4模型

- Python3 phi4_potential.py
- 产生本PPT中ϕ^4模型的相关图形：



练习 2：一维示例

- Python3 1d_potential.py
- 产生本PPT有关$V(\phi, T) = (cT^2 - m^2)\phi^2 + \kappa\phi^3 + \lambda\phi^4$的图形：



练习 3：安装PhaseTracer

You need a C++14 compliant compiler and our dependencies. The dependencies can be installed by

Ubuntu/Debian

```
sudo apt install libalglib-dev libnlopt-cxx-dev libeigen3-dev libboost-filesystem-dev li
```

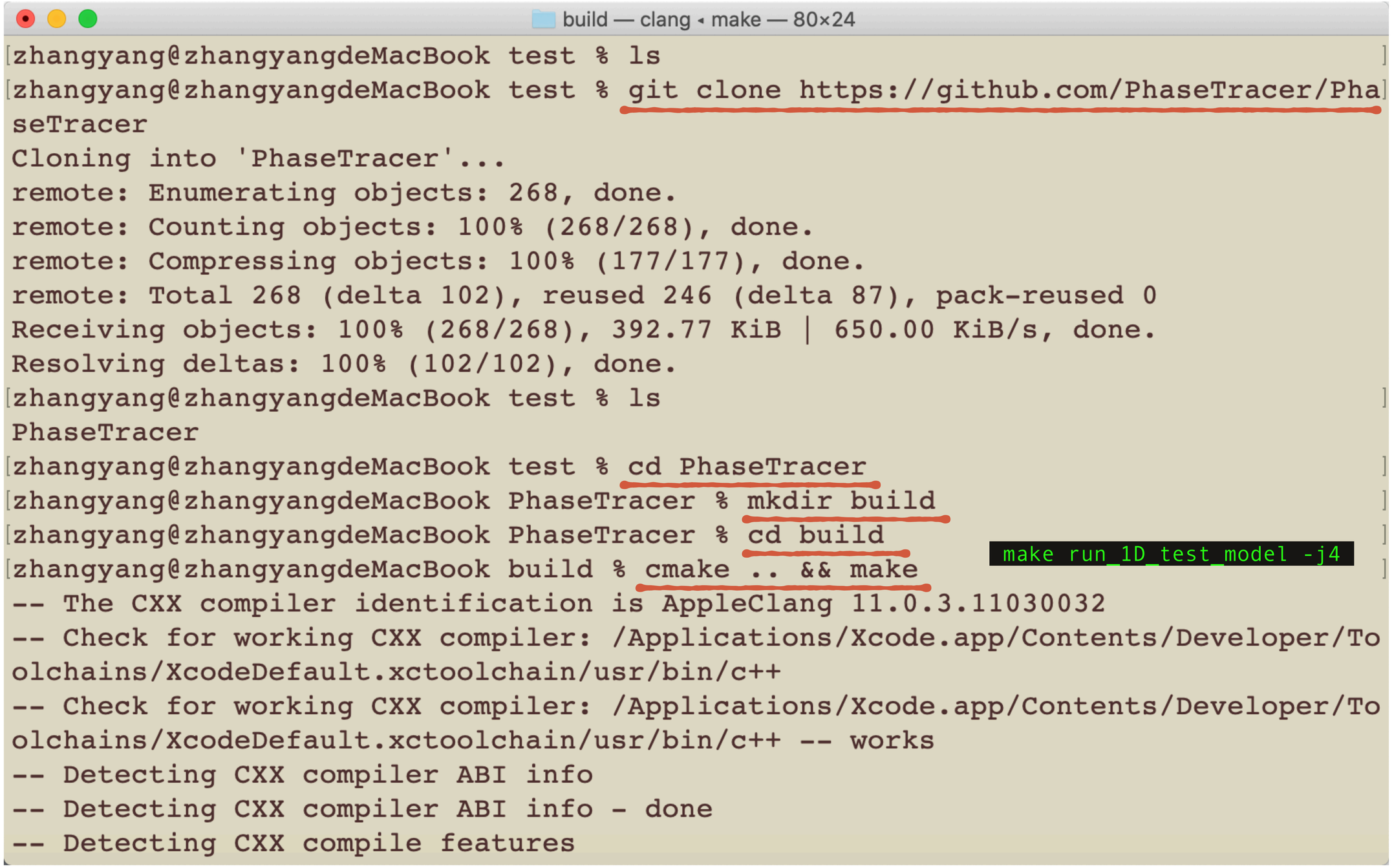
Fedora

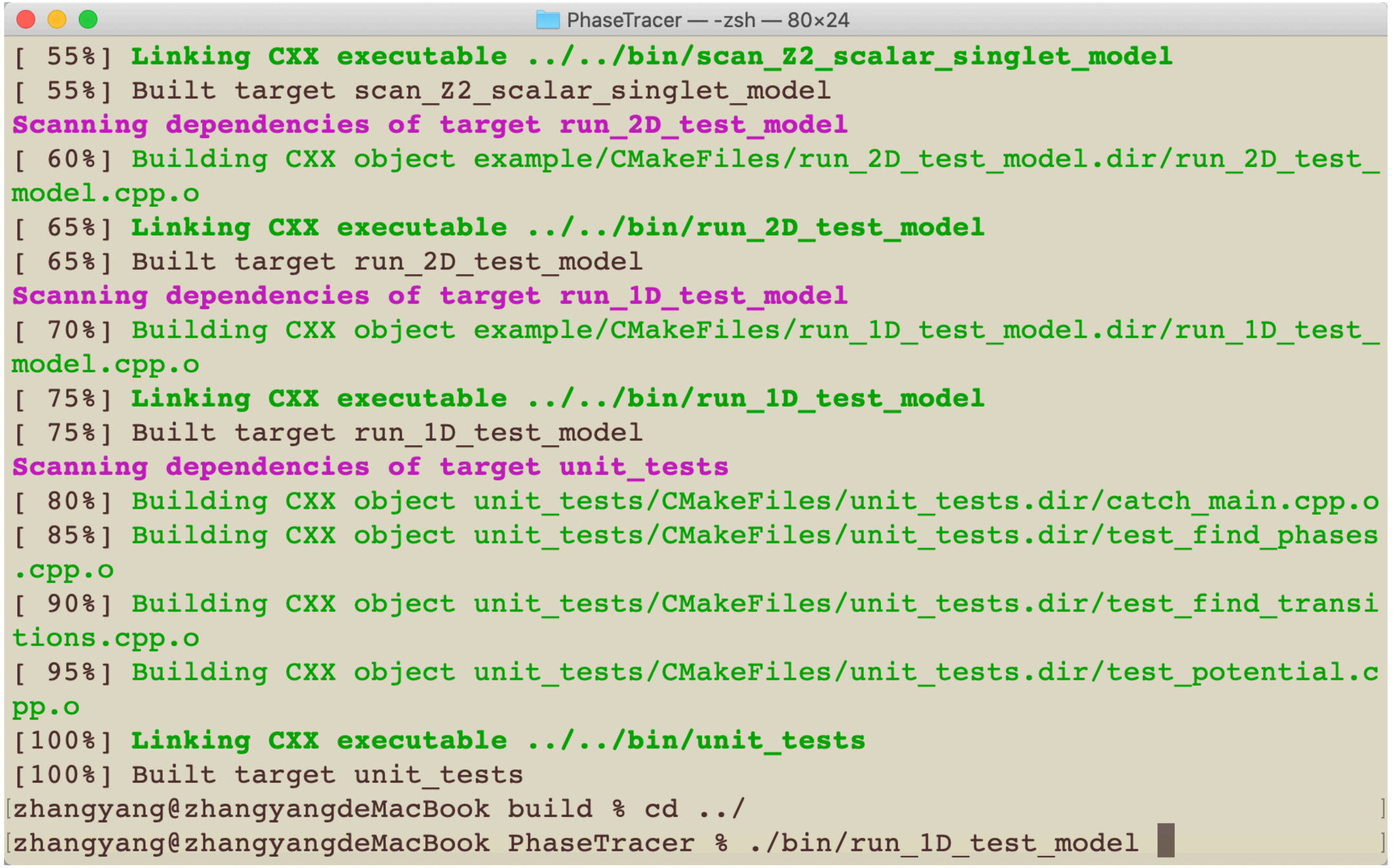
```
sudo dnf install alglib-devel nlopt-devel eigen3-devel boost-devel gsl-devel
```

Mac

```
brew install alglib nlopt eigen boost gsl
```

If alglib is not found, see https://github.com/S-Dafarra/alglib-cmake

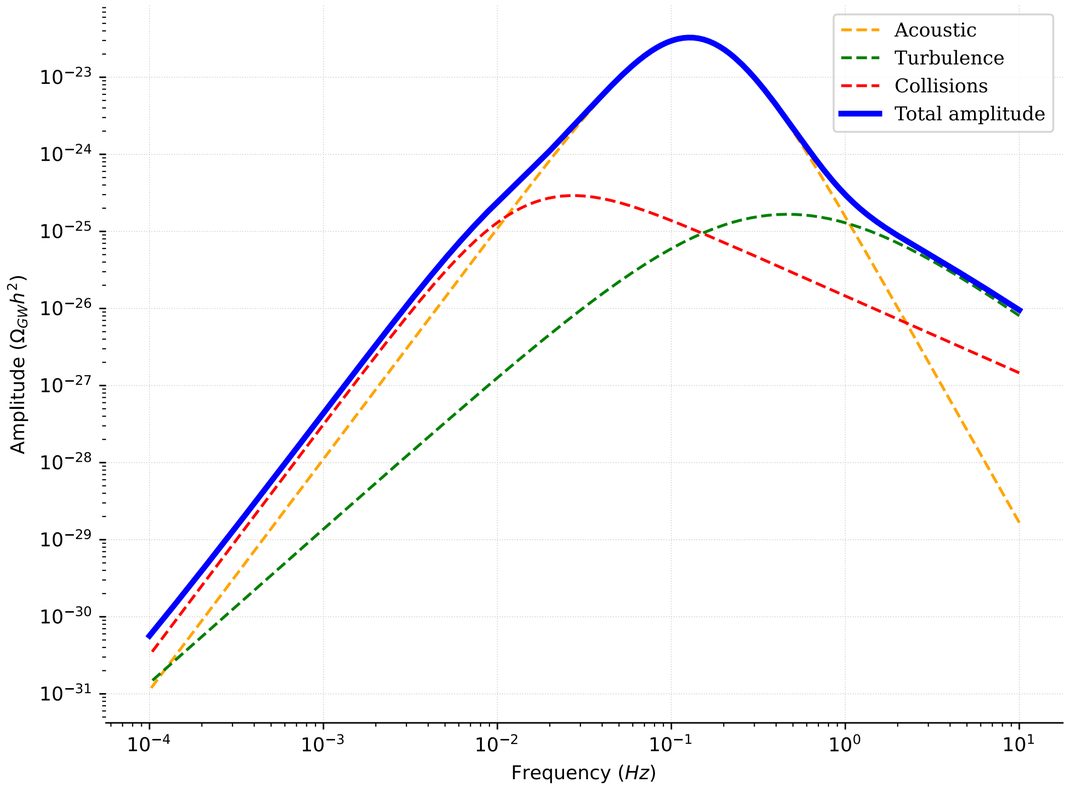

build — clang ◂ make — 80×24
zhangyang@zhangyangdeMacBook test % ls
zhangyang@zhangyangdeMacBook test % git clone https://github.com/PhaseTracer/PhaseTracer
Cloning into 'PhaseTracer'...
remote: Enumerating objects: 268, done.
remote: Counting objects: 100% (268/268), done.
remote: Compressing objects: 100% (177/177), done.
remote: Total 268 (delta 102), reused 246 (delta 87), pack-reused 0
Receiving objects: 100% (268/268), 392.77 KiB | 650.00 KiB/s, done.
Resolving deltas: 100% (102/102), done.
zhangyang@zhangyangdeMacBook test % ls
PhaseTracer
zhangyang@zhangyangdeMacBook test % cd PhaseTracer
zhangyang@zhangyangdeMacBook PhaseTracer % mkdir build
zhangyang@zhangyangdeMacBook PhaseTracer % cd build
zhangyang@zhangyangdeMacBook build % cmake .. && make
make run_1D_test_model -j4
-- The CXX compiler identification is AppleClang 11.0.3.11030032
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features


PhaseTracer — -zsh — 80×24
[55%] Linking CXX executable ../../bin/scan_Z2_scalar_singlet_model
[55%] Built target scan_Z2_scalar_singlet_model
Scanning dependencies of target run_2D_test_model
[60%] Building CXX object example/CMakeFiles/run_2D_test_model.dir/run_2D_test_model.cpp.o
[65%] Linking CXX executable ../../bin/run_2D_test_model
[65%] Built target run_2D_test_model
Scanning dependencies of target run_1D_test_model
[70%] Building CXX object example/CMakeFiles/run_1D_test_model.dir/run_1D_test_model.cpp.o
[75%] Linking CXX executable ../../bin/run_1D_test_model
[75%] Built target run_1D_test_model
Scanning dependencies of target unit_tests
[80%] Building CXX object unit_tests/CMakeFiles/unit_tests.dir/catch_main.cpp.o
[85%] Building CXX object unit_tests/CMakeFiles/unit_tests.dir/test_find_phases.cpp.o
[90%] Building CXX object unit_tests/CMakeFiles/unit_tests.dir/test_find_transitions.cpp.o
[95%] Building CXX object unit_tests/CMakeFiles/unit_tests.dir/test_potential.cpp.o
[100%] Linking CXX executable ../../bin/unit_tests
[100%] Built target unit_tests
[zhangyang@zhangyangdeMacBook build % cd ../
[zhangyang@zhangyangdeMacBook PhaseTracer % ./bin/run_1D_test_model

PhaseTracer — -zsh — 68×46

```
[zy@Yangs-MacBook-Pro-2 PhaseTracer % ./bin/run_1D_test_model
found 2 phases

=== phase key = 0 ===
Maximum temperature = 1000
Minimum temperature = 33.1513
Field at tmax = [4.80762e-06]
Field at tmin = [-1.52561e-05]
Potential at tmax = 2.30901e-06
Potential at tmin = 2.30449e-09
Ended at tmax = Reached tstop
Ended at tmin = Jump in fields indicated end of phase

=== phase key = 1 ===
Maximum temperature = 61.7437
Minimum temperature = 0
Field at tmax = [37.8322]
Field at tmin = [81.1597]
Potential at tmax = 65886.6
Potential at tmin = -1.66587e+06
Ended at tmax = Jump in fields indicated end of phase
Ended at tmin = Reached tstop

found 1 transition

did not check for subcritical transitions

=== transition from phase 0 to phase 1 ===
changed = [true]
TC = 59.1608
false vacuum (TC) = [4.36255e-06]
true vacuum (TC) = [50.0002]
gamma (TC) = 0.845158
delta potential (TC) = 0.00117793
TN = 57.4083
false vacuum (TN) = [4.89914e-06]
true vacuum (TN) = [53.5749]
transition was not subcritical

=== gravitational wave spectrum generated at T = 57.4083 ===
alpha = 0.00138403
beta over H = 7411.19
peak frequency = 0.127706
peak amplitude = 3.28843e-23
signal to noise ratio for LISA = 6.24943e-13
zy@Yangs-MacBook-Pro-2 PhaseTracer %
```

```
% ./bin/run_1D_test_model

% ./bin/run_1D_test_model -d
```



练习4：运行PhaseTracer中的一维示例

PhaseTracer / EffectivePotential / include / models / **1D_test_model.hpp**

$$V(\phi, T) = (cT^2 - m^2)\phi^2 + \kappa\phi^3 + \lambda\phi^4$$

```
#include "potential.hpp"
#include "pow.hpp"

namespace EffectivePotential {
// Publicy inherit from the EffectivePotential class
class OneDimModel : public Potential {
  public:
    // Implement our scalar potential - this is compulsory
    double V(Eigen::VectorXd x, double T) const override {
      return (0.1 * square(T) - square(100.)) * square(x[0])
             - 10. * cube(x[0]) + 0.1 * pow_4(x[0]);
    }
    // Declare the number of scalars in this model - this is compulsory
    size_t get_n_scalars() const override { return 1; }
    // Look at x >= 0 - this is optional
    bool forbidden(Eigen::VectorXd x) const override { return x[0] < -0.1;}
};
}
```

练习4：运行PhaseTracer中的一维示例

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

$$V(\phi, T) = (0.1T^2 - 100)\phi^2 - 10\phi^3 + 0.1\phi^4$$

PhaseTracer — -zsh — 76×42

```
=== phase key = 0 ===
Maximum temperature = 1000
Minimum temperature = 33.1513
Field at tmax = [9.2114e-06]
Field at tmin = [-1.52352e-05]
Potential at tmax = 8.4765e-06
Potential at tmin = 2.29817e-09
Ended at tmax = Reached tstop
Ended at tmin = Jump in fields indicated end of phase

=== phase key = 1 ===
Maximum temperature = 61.7437
Minimum temperature = 0
Field at tmax = [37.8322]
Field at tmin = [81.1597]
Potential at tmax = 65886.6
Potential at tmin = -1.66587e+06
Ended at tmax = Jump in fields indicated end of phase
Ended at tmin = Reached tstop

[2020-03-30 11:56:37.364765] [0x0000000104d0ddc0] [debug]   Finding critical
 temperatures between phases 0 and 1
[2020-03-30 11:56:37.365935] [0x0000000104d0ddc0] [debug]   Found critical t
emperature = 59.1608
found 1 transition

=== transition from phase 0 to phase 1 ===
false vacuum = [7.66464e-06]
true vacuum = [50.0002]
changed = [true]
TC = 59.1608
gamma = 0.845158
delta potential = 0.00117793

[2020-03-30 11:56:37.367426] [0x0000000104d0ddc0] [debug]   Executing python
 -W ignore /Users/zhangyang/work/test/PhaseTracer/include/../make_plots/phas
e_plotter.py 1D_test_model
Making plots with 1 fields
Plotting phases against temperature for phi_T_1D_test_model.pdf figure
Plotting potential against temperature for V_T_1D_test_model.pdf figure
zhangyang@zhangyangdeMacBook PhaseTracer %
```

$$T_C = \sqrt{\frac{\kappa^2 + 4\lambda m^2}{4c\lambda}} = 59.1608$$

PhaseTracer / example / **run_1D_test_model.cpp**

```cpp
#include <iostream>
#include "models/1D_test_model.hpp"
#include "phase_finder.hpp"
#include "transition_finder.hpp"

int main() {
  EffectivePotential::OneDimModel model;  // Construct

  PhaseTracer::PhaseFinder pf(model);  // Construct th
  pf.find_phases();  // Find the phases
  std::cout << pf;  // Print information about the pha

  PhaseTracer::TransitionFinder tf(pf);  // Construct
  tf.find_transitions();  // Find the transitions
  std::cout << tf;  // Print information about the tra
  return 0;
}
```

练习5：扫描PhaseTracer中的一维示例

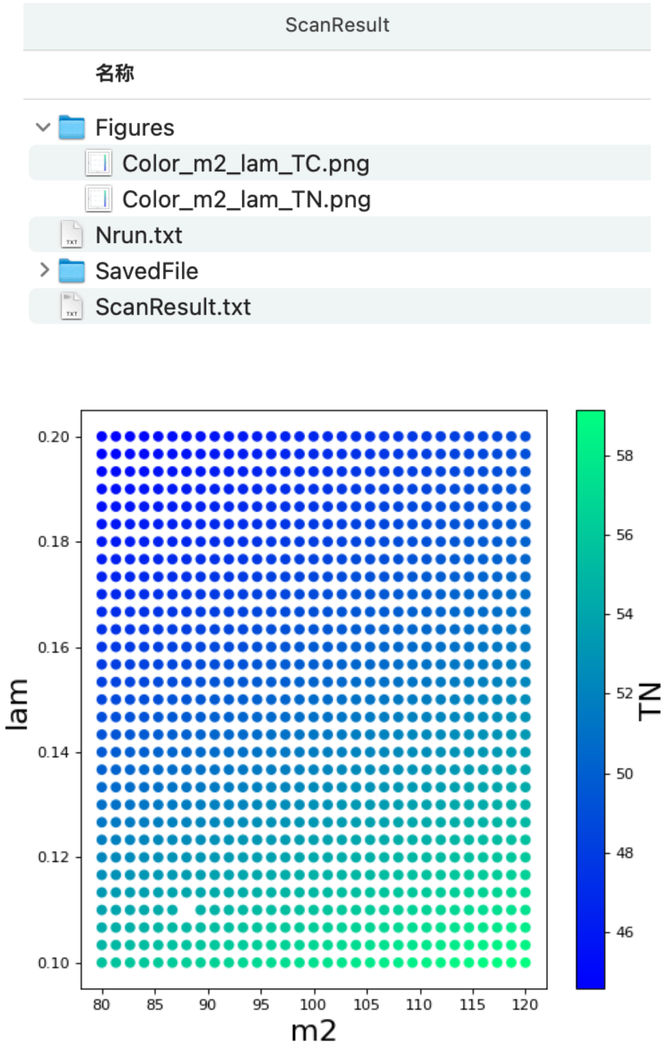
河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

```
% ./EasyScan_HEP/bin/easyscan.py grid_scan.ini
```

Program summary

Program Title: `EasyScan_HEP`
CPC Library link to program files: https://doi.org/10.17632/4fcb77dxfw.1
Developer's repository link: https://github.com/phyzhangyang/EasyScan_HEP
Licensing provisions: Apache 2.0
Programming language: Python
Nature of problem: Performing numerical analysis of new physics models is crucial in High Energy Physics (HEP), and requires scanning parameter space using various HEP packages. Connecting these packages together can be cumbersome, time-consuming, and prone to errors, especially when using advanced scanning methods, due to the lack of a unified interface between these software and scanning methods.
Solution method: `EasyScan_HEP` utilizes the `ConfigParser` module in Python to read a unified and human-readable configuration file that connects HEP packages and sets scanning methods. We employ the `subprocess.Popen` and `os.system` functions to execute HEP programs, and provide users with various options to configure input parameters and retrieve output parameters, as well as several pre-installed scanning methods.
Additional comments including restrictions and unusual features: `EasyScan_HEP` is not designed for specific models or HEP packages. Instead, it is compatible with almost any program that can be executed via the command line, requiring minimal interface modifications.

arXiv:2304.03636



练习5：扫描PhaseTracer中的一维示例

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

grid_scan.ini

```
[scan]
Result folder name:  ScanResult
Scan method:         grid
#                    varID  Prior   Min      MAX     BinNum
Input parameters:    m2,     flat,   80,      120,    30
                     lam,    flat,   0.1,     0.2,    30
Parallel threads:    4
Parallel folder:     PhaseTracer

[program1]
Program name:     run_1D_test_model
Execute command: ./bin/run_1D_test_model  -f >log.txt
Command path:     PhaseTracer/
Input file:       1, PhaseTracer/input.txt
#                 varID    fileID  method
Input variable:   m2,        1,      Position, 1,  1
                  lam,       1,      Position, 1,  2
Output file:      1, PhaseTracer/output.txt
#                 varID    fileID  method
Output variable: TC,        1,      Position, 1,  1
                  TN,       1,      Position, 1,  2


[plot]
#       x-axis    y-axis     color     FigureName
Color: m2,        lam,         TC
       m2,        lam,         TN
```

ScanResult.txt

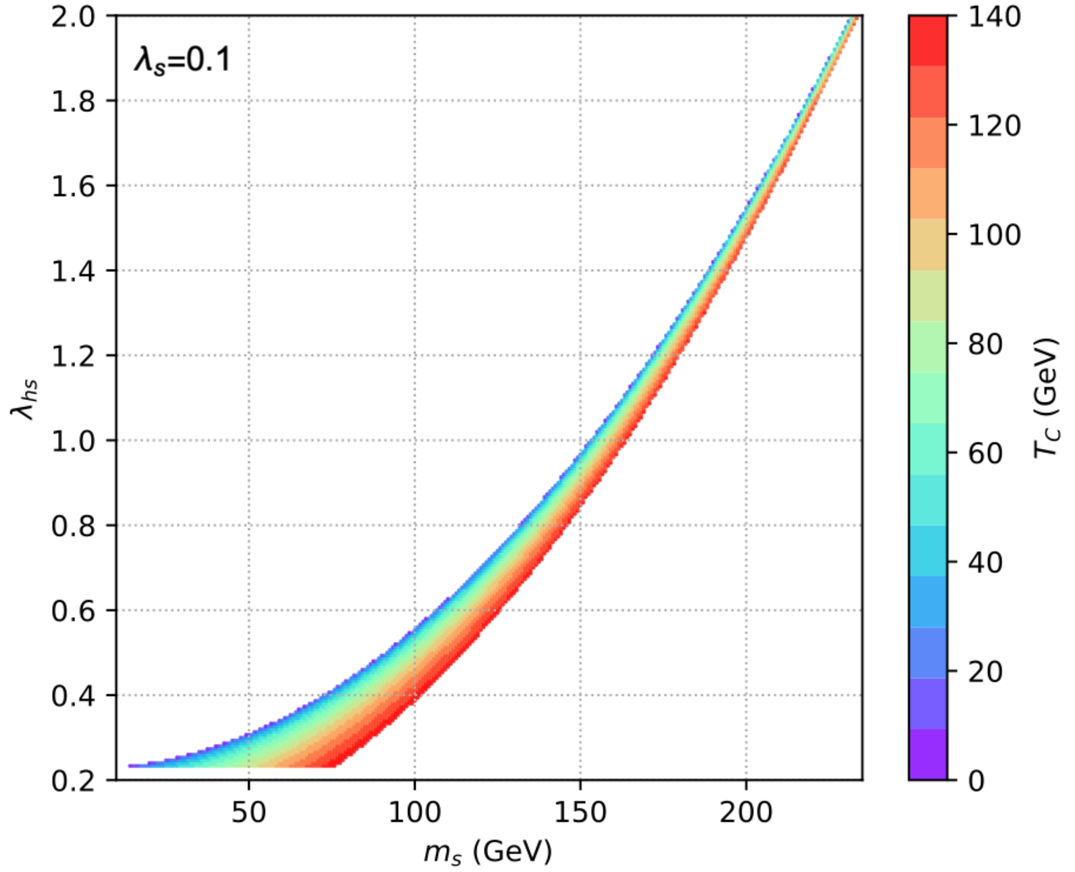
```
lam,m2,TC,TN,Chi2
0.1,80.0,57.4456,55.6267,0.0
0.10333333333333333,80.0,56.7394,54.9773,0.0
0.18000000000000002,89.33333333333333,47.7726,46.9649,0.0
0.10666666666666667,80.0,56.0692,54.3824,0.0
0.18333333333333335,89.33333333333333,47.5076,46.7195,0.0
0.15333333333333335,100.0,51.2878,50.3061,0.0
0.11000000000000001,80.0,55.4322,53.822,0.0
0.18666666666666668,89.33333333333333,47.2506,46.4748,0.0
0.15666666666666668,100.0,50.9485,50.0163,0.0
0.11333333333333334,80.0,54.8259,53.268,0.0
0.19,89.33333333333333,47.0013,46.2532,0.0
0.16,100.0,50.6211,49.7025,0.0
0.11666666666666667,80.0,54.2481,52.748,0.0
0.19333333333333336,89.33333333333333,46.7593,46.0192,0.0
0.16333333333333333,100.0,50.3052,49.4107,0.0
0.12000000000000001,80.0,53.6967,52.2458,0.0
0.19666666666666668,89.33333333333333,46.5244,45.8057,0.0
0.16666666666666669,100.0,50,49.1325,0.0
0.12333333333333334,80.0,53.1698,51.7467,0.0
0.2,89.33333333333333,46.2961,45.6002,0.0
0.16999999999999998,100.0,49.705,48.8603,0.0
0.12666666666666668,80.0,52.6658,51.3294,0.0
0.12666666666666668,110.66666666666666,55.5009,54.2653,0.0
0.1,90.66666666666667,58.3667,56.5643,0.0
0.17333333333333334,100.0,49.4197,48.5973,0.0
0.13,80.0,52.1831,50.8826,0.0
0.13,110.66666666666666,55.0431,53.8559,0.0
0.10333333333333333,90.66666666666667,57.6717,55.9549,0.0
0.17666666666666667,100.0,49.1436,48.3443,0.0
```

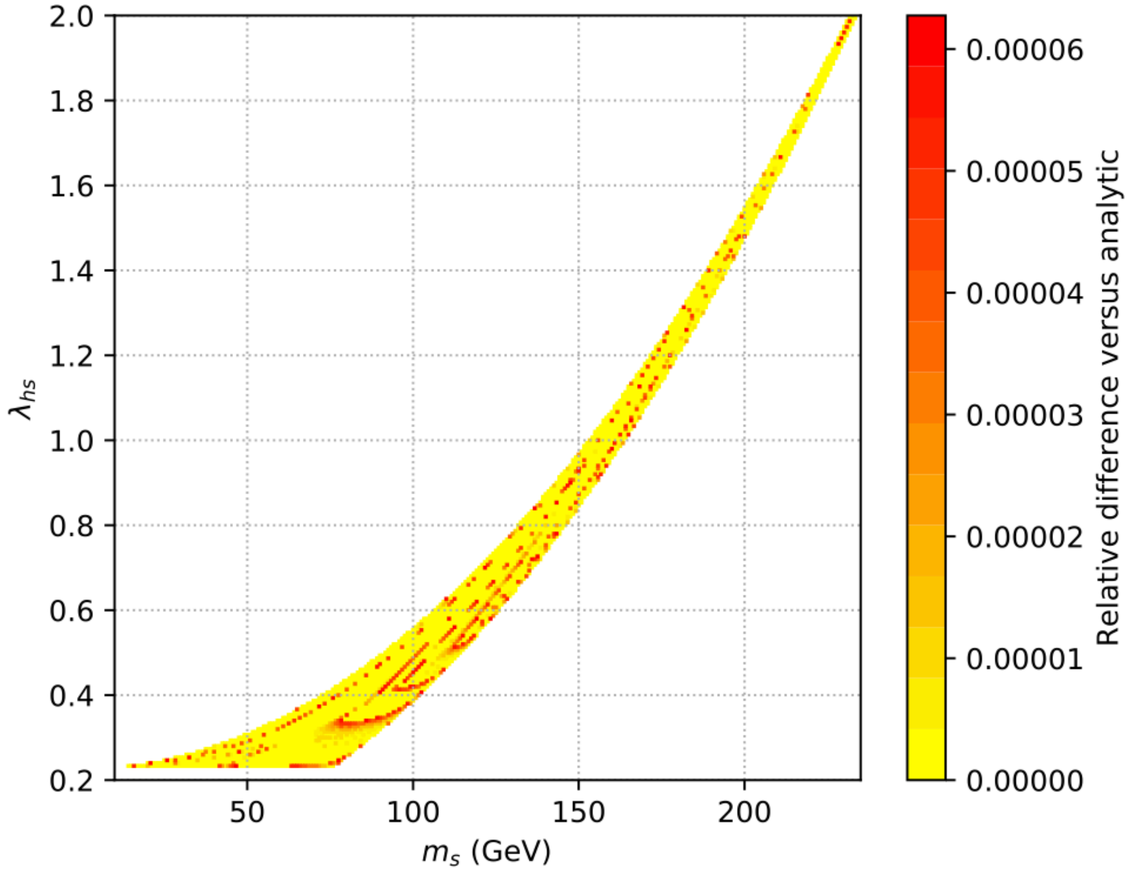
练习6：添加 Z2 Scalar Singlet Model

$$V(H,s) = \mu_{\rm h}^2 H^\dagger H + \lambda_{\rm h}(H^\dagger H)^2 + \frac{\lambda_{\rm hs}}{2}(H^\dagger H)s^2 + \frac{\mu_{\rm s}^2}{2}s^2 + \frac{\lambda_{\rm s}}{4}s^4.$$

$$\mu_{\rm s}(T)^2 = \mu_{\rm s}^2 + c_{\rm s}T^2, \quad \mu_{\rm h}(T)^2 = \mu_{\rm h}^2 + c_{\rm h}T^2$$

$$T_c^2 = \frac{\lambda_{\rm h}c_{\rm s}\mu_{\rm s}^2 - \lambda_{\rm s}c_{\rm h}\mu_{\rm h}^2 - \sqrt{\lambda_{\rm h}\lambda_{\rm s}}|c_{\rm s}\mu_{\rm h}^2 - c_{\rm h}\mu_{\rm s}^2|}{\lambda_{\rm s}c_{\rm h}^2 - \lambda_{\rm h}c_{\rm s}^2}$$

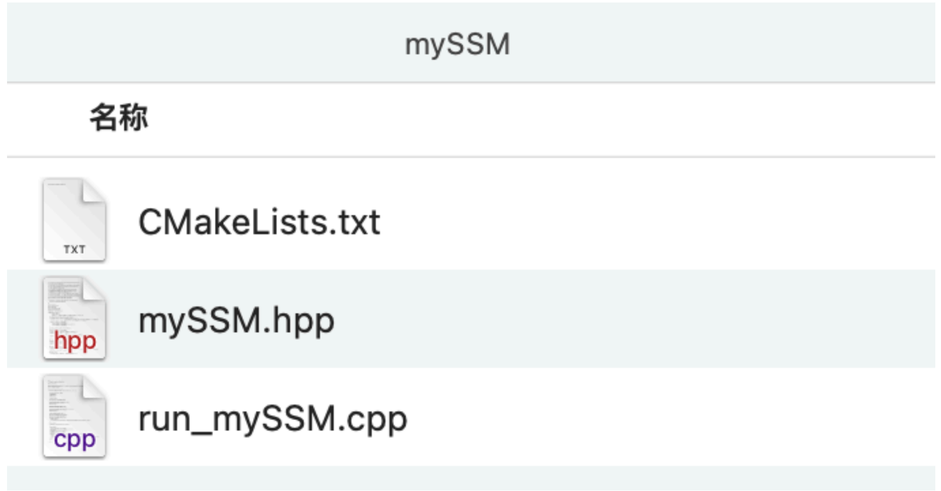




练习6：添加 Z2 Scalar Singlet Model

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

- 在PhaseTracer/example中创建mySSM文件夹
- 在mySSM文件夹添加



CMakeLists.txt

```
add_executable(run_mySSM run_mySSM.cpp)
```

- run_mySSM.cpp 与模型无关，大同小异，根据需求改设置，增加输入输出

run_mySSM.cpp

```
/**
  My SSM example
*/
#include <iostream>
#include "mySSM.hpp"
#include "phasetracer.hpp"

int main(int argc, char* argv[]) {
  // Construct model
  EffectivePotential::mySSM model;
  // Make PhaseFinder object and find the phases
  PhaseTracer::PhaseFinder pf(model);
  pf.find_phases();
  std::cout << pf;
  // Make ActionCalculator object
  PhaseTracer::ActionCalculator ac(model);
  // Make TransitionFinder object and find the transitions
  PhaseTracer::TransitionFinder tf(pf,ac);
  tf.find_transitions();
  std::cout << tf;
  return 0;
}
```

练习6：添加 Z2 Scalar Singlet Model

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

仿照 1D_test_model.hpp 编写 mySSM.hpp

```cpp
#ifndef POTENTIAL_MYSSM_HPP_INCLUDED
#define POTENTIAL_MYSSM_HPP_INCLUDED
#include <vector>
#include "potential.hpp"
#include "pow.hpp"
namespace EffectivePotential {
class mySSM : public Potential {
public:
  double V(Eigen::VectorXd phi, double T) const override {
    const double mus_sq = square(m_s) - lambda_hs * square(v) / 2.;
    const double muh_sq_T = muh_sq + square(T) / 48. * (9. * g_sq + 3. * gp_sq + 12. * yt_sq + 24. * lambda_h + 2. * lambda_hs);
    const double mus_sq_T = mus_sq + square(T) / 12. * (2. * lambda_hs + 3. * lambda_s);
    return 0.5 * muh_sq_T * square(phi[0]) + 0.25 * lambda_h * pow_4(phi[0]) + 0.25 * lambda_hs * square(phi[0]) * square(phi[1]) +
           0.5 * mus_sq_T * square(phi[1]) +0.25 * lambda_s * pow_4(phi[1]);
  }
  size_t get_n_scalars() const override { return 2; }
  std::vector<Eigen::VectorXd> apply_symmetry(Eigen::VectorXd phi) const override {
    auto phi1 = phi; phi1[0] = -phi[0]; auto phi2 = phi; phi2[1] = -phi[1];
    return {phi1, phi2};
  };
private:
  const double v = 247.4544243292407; const double mh = 125.25;
  const double g = 0.6477096097526751; const double g_sq = g * g;
  const double gp = 0.3585644903737741; const double gp_sq = gp * gp;
  const double yt_sq = 0.9341361105006658;
  const double muh_sq = -0.5 * mh * mh; const double lambda_h = -muh_sq / square(v);
  double lambda_hs = 0.3; double m_s = 65; double lambda_s = 0.1;
};
} // namespace EffectivePotential
#endif
```

$$\mu_s(T)^2 = \mu_s^2 + c_s T^2, \quad \mu_h(T)^2 = \mu_h^2 + c_h T^2$$

$$V(H,s) = \mu_h^2 H^\dagger H + \lambda_h (H^\dagger H)^2 + \frac{\lambda_{hs}}{2}(H^\dagger H)s^2 + \frac{\mu_s^2}{2}s^2 + \frac{\lambda_s}{4}s^4.$$

$$V(\phi_1, \phi_2) = V(\phi_1, -\phi_2) = V(-\phi_1, \phi_2)$$

$$\left.\frac{\partial^2 V}{\partial \phi_h^2}\right|_{\boldsymbol{v}} = m_h^2 \quad \text{and} \quad \left.\frac{\partial^2 V}{\partial \phi_s^2}\right|_{\boldsymbol{v}} = m_s^2.$$

$$\overline{m}_h^2 = -\mu_h^2 + 3\lambda_h v_h^2 = 2\lambda_h v_h^2,$$

$$\overline{m}_s^2 = -\mu_s^2 + \frac{\lambda_{hs}}{2} v_h^2.$$

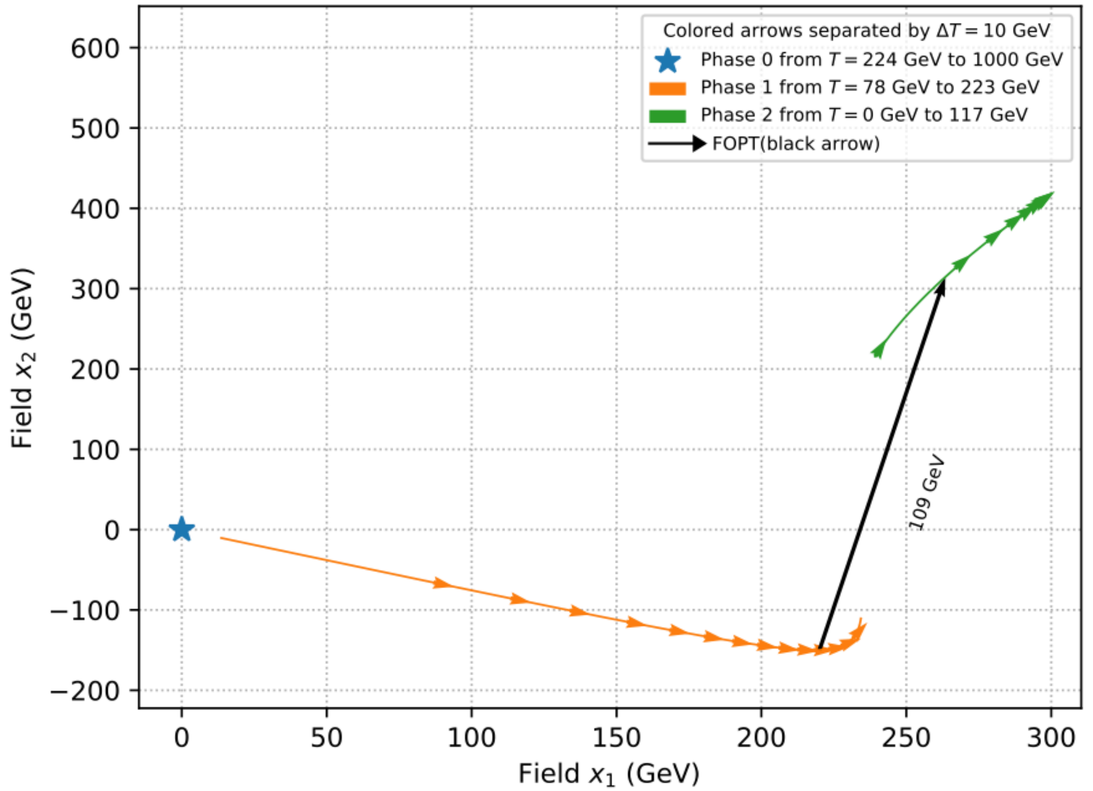
练习7：运行PhaseTracer中的二维示例

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University

```
% ./bin/run_2D_test_model
```

PhaseTracer / example / **run_2D_test_model.cpp**

```
27    // Construct our model
28    EffectivePotential::TwoDimModel model;
29
30    // Make PhaseFinder object and find the phases
31    PhaseTracer::PhaseFinder pf(model);
32    pf.find_phases();
33    std::cout << pf;
34
35    // Make TransitionFinder object and find the transitions
36    PhaseTracer::TransitionFinder tf(pf);
37    tf.find_transitions();
38    std::cout << std::setprecision (15) << tf;
39
40    if (debug_mode) {
41      PhaseTracer::potential_plotter(model, tf.get_transitions().front().TC, "2D_test_model", 0., 2., 0.01, -2., 0., 0.01);
42      PhaseTracer::potential_line_plotter(model, tf.get_transitions(), "2D_test_model");
43      PhaseTracer::phase_plotter(tf, "2D_test_model");
44    }
```



练习7：运行PhaseTracer中的二维示例

- The effective potential typically includes the following contributions,

$$V_{\text{eff}} = V^{\text{tree}} + \Delta V^{\text{1-loop}} + \Delta V_T^{\text{1-loop}} + V_{\text{daisy}}.$$

$$\begin{aligned}\Delta V^{\text{1-loop}} = \frac{1}{64\pi^2}\Bigg(&\sum_\phi n_\phi m_\phi^4(\xi)\left[\ln\left(\frac{m_\phi^2(\xi)}{Q^2}\right) - 3/2\right] \\ &+ \sum_V n_V m_V^4\left[\ln\left(\frac{m_V^2}{Q^2}\right) - 5/6\right] \\ &- \sum_V \tfrac{1}{3} n_V (\xi m_V^2)^2\left[\ln\left(\frac{\xi m_V^2}{Q^2}\right) - 3/2\right] \\ &- \sum_f n_f m_f^4\left[\ln\left(\frac{m_f^2}{Q^2}\right) - 3/2\right]\Bigg).\end{aligned}$$

$$\begin{aligned}\Delta V_T^{\text{1-loop}} = \frac{T^4}{2\pi^2}\Bigg[&\sum_\phi n_\phi J_{\text{B}}\left(\frac{m_\phi^2(\xi)}{T^2}\right) + \sum_V n_V J_{\text{B}}\left(\frac{m_V^2}{T^2}\right) \\ &- \sum_V \tfrac{1}{3} n_V J_{\text{B}}\left(\frac{\xi m_V^2}{T^2}\right) + \sum_f n_f J_{\text{F}}\left(\frac{m_f^2}{T^2}\right)\Bigg],\end{aligned}$$

$$J_{\text{B/F}}(y^2) = \pm\,\text{Re}\int_0^\infty x^2 \ln\left(1 \mp e^{-\sqrt{x^2+y^2}}\right) dx.$$

$$V_{\text{daisy}} = -\frac{T}{12\pi}\left(\sum_\phi n_\phi\left[\left(\overline{m_\phi^2}\right)^{\frac{3}{2}} - \left(m_\phi^2\right)^{\frac{3}{2}}\right] + \sum_V \tfrac{1}{3} n_V\left[\left(\overline{m_V^2}\right)^{\frac{3}{2}} - \left(m_V^2\right)^{\frac{3}{2}}\right]\right),$$

练习7：运行PhaseTracer中的二维示例

河南师范大学 理论物理中心
Centre for Theoretical Physics
Henan Normal University

PhaseTracer / EffectivePotential / include / models / **2D_test_model.hpp**

```
33  namespace EffectivePotential {
34
35  class TwoDimModel : public OneLoopPotential {
36   public:
37
38    double V0(Eigen::VectorXd phi) const override {
39      return 0.25 * l1 * square(square(phi[0]) - square(v))
40             + 0.25 * l2 * square(square(phi[1]) - square(v))
41             - square(mu) * phi[0] * phi[1];
42    }
43
44    std::vector<double> get_scalar_masses_sq(Eigen::VectorXd phi, double xi) const override {
45      const double a = l1 * (3. * square(phi[0]) - square(v));
46      const double b = l2 * (3. * square(phi[1]) - square(v));
47      const double A = 0.5 * (a + b);
48      const double B = std::sqrt(0.25 * square(a - b) + pow_4(mu));
49      const double mb_sq = y1 * (square(phi[0]) + square(phi[1])) + y2 * phi[0] * phi[1];
50      return {A + B, A - B,mb_sq};
51    }
52
53    std::vector<double> get_scalar_dofs() const override { return {1., 1., 30.}; }
54    size_t get_n_scalars() const override { return 2; }
55
56    std::vector<Eigen::VectorXd> apply_symmetry(Eigen::VectorXd phi) const override {
57      return {-phi};
58    };
```

$$V(\phi_1,\phi_2)=\frac{1}{8}\frac{m_1^2}{v^2}(\phi_1^2-v^2)^2+\frac{1}{8}\frac{m_2^2}{v^2}(\phi_2^2-v^2)^2-\mu^2\phi_1\phi_2.$$

$$M^2(\phi_1,\phi_2)=\begin{pmatrix}\frac{m_1^2}{2v^2}(3\phi_1^2-v^2) & -\mu^2\\ -\mu^2 & \frac{m_2^2}{2v^2}(3\phi_2^2-v^2)\end{pmatrix}.$$

$$m_X(\phi_1,\phi_2)=y_A^2(\phi_1^2+\phi_2^2)+y_b^2\phi_1\phi_2$$

$$n_X=30$$

总结

河南师范大学
理论物理中心
Centre for Theoretical Physics
Henan Normal University