SRI VENKATESWARA INTERNSHIP PROGRAM FOR RESEARCH IN ACADEMICS (SRI-VIPRA)

SRI-VIPRA

Project Report of 2024: SVP-2464

"Studying BSM Physics with SARAH"

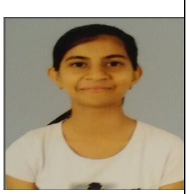
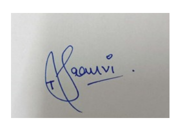
IQAC
Sri Venkateswara College
University of Delhi
Benito Juarez Road, Dhaula Kuan, New Delhi
New Delhi -110021

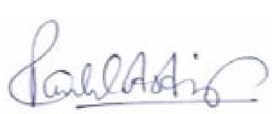
SRIVIPRA PROJECT 2024

Title : Studying BSM Physics with SARAH (SVP – 2464)

Name of Mentor: Ram Lal Awasthi **Name of Department: Physics** **Designation: Assistant Professor**	**Photo:** 

List of students under the SRIVIPRA Project

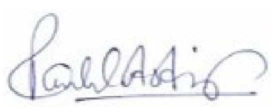
S.No	**Photo**	**Name of the student**	**Roll number**	**Course**	**Signature**
1		Mahi Garg	1822021	B.Sc. (H) Physics	
2		Saanvi Tiwari	1822047	B.Sc. (H) Physics	



Signature of Mentor

Certificate of Originality

This is to certify that the aforementioned students from Sri Venkateswara College have participated in the summer project SVP-2464 titled ‘Studying BSM Physics with SARAH’. The participants have carried out the research project work under my guidance and supervision from 1st July, 2024 to 30th September 2024. The work carried out is original and carried out in an hybrid mode.



Signature of Mentor

SRI VENKATESWARA INTERNSHIP PROGRAM FOR RESEARCH IN ACADEMICS (SRI-VIPRA'24)

Project Report

Studying BSM Physics with SARAH (SVP-2464)

Mahi Garg (1822021) and Saanvi Tiwari (1822047)
BSc(Hons)Physics (3rd Year)

IQAC
Sri Venkateswara College
University of Delhi
Benito Juarez Road, Dhaula Kuan, New Delhi New Delhi -110021

Certificate of Originality

This is to certify that students Ms. Mahi Garg and Ms. Saanvi Tiwari from Sri Venkateswara College has participated in the summer project SVP-2464 titled "Computation of Certain Processes in Particle Physics". They have carried out the research project work under my guidance and supervision from 1st July, 2024 to 30th September 2024. The work carried out is original learning project and carried out in an online/offline/hybrid mode.

Signature of Mentor

Acknowledgement

We would like to express our sincere gratitude to Sri Venkateswara College, University of Delhi, for offering us the research opportunity under the Sri Venkateswara Internship Program in Research and Academics (SRI-VIPRA). We are immensely thankful for providing the essential resources and a conducive environment that enabled the successful completion of this project.

We would like to express our deepest gratitude to Prof. Ram Lal Awasthi, for his invaluable guidance, support, and encouragement throughout our internship. His expertise and insightful feedback have been instrumental in helping us understand complex concepts and successfully complete this project. We greatly appreciate his patience and willingness to share knowledge, which has significantly contributed to our personal and professional growth.

Mahi Garg and Saanvi Tiwari

Contents

1 Introduction

This report presents a comprehensive account of the research process, methodologies employed, and the outcomes of our project, which focuses on concepts in physics beyond the Standard Model.

We began by studying the Standard Model, a theoretical framework that describes the fundamental particles and forces responsible for various phenomena in the universe. This model categorizes all known elementary particles into quarks, leptons, and force carriers such as photons, W and Z bosons, and gluons. It also includes the Higgs boson, which is essential for giving particles mass. The Standard Model successfully explains electromagnetic, weak, and strong interactions, forming the cornerstone of particle physics.

The primary aim of our project was to explore specialized mathematical tools, particularly SARAH—a Mathematica software package widely used in particle physics and dark matter research. SARAH facilitates the construction and analysis of models extending beyond the Standard Model. Through this tool, we delved into the theoretical frameworks surrounding dark matter and its interactions, generating the necessary input files for MicrOmegas, a computational tool designed for dark matter analysis. Working with SARAH provided us with valuable hands-on experience in building and analyzing models crucial for advanced research in particle physics.

Additionally, we gained a brief but meaningful introduction to quantum field theory (QFT), a fundamental framework describing how fields, such as the electromagnetic field, interact with particles. While our engagement with QFT was limited, it laid the groundwork for future exploration into the deeper aspects of particle physics.

Overall, this project has successfully bridged computational skills with advanced theoretical concepts, positioning us well for further investigations in both particle physics and computational mathematics.

2 Methodology

This section outlines the methodologies employed to achieve the goals of our project, which centered on gaining a comprehensive understanding of SARAH—a powerful Mathematica software package used in particle physics. Our approach involved a systematic exploration of the software, allowing us to delve into model definition and analysis that extends beyond the Standard Model.

To effectively navigate this process, we engaged with the relevant documentation and tutorials, which provided the foundational knowledge necessary for leveraging SARAH's functionalities. This methodological framework combined both theoretical exploration and practical application, ultimately positioning us to analyze models relevant to dark matter research.

2.1 Objectives

The primary objectives of this project were as follows:

- To install and set up SARAH in our systems.
- To define a new model within the software that extends beyond the Standard Model.
- To analyze the defined model to extract meaningful insights related to particle physics.

In the Following subsections we will explore the Mathematica Package- SARAH, study its properties and use to get important analytical information.

2.2 SARAH: A Mathematica Package

SARAH is a Mathematica package for building and analyzing particle physics models, supporting both supersymmetric and non-supersymmetric models. It generates detailed outputs such as vertices, mass matrices, tadpole equations, and renormalization group equations (RGEs), which can be exported to LaTeX for PDF compilation.

Additionally, SARAH creates input files for various tools, including FeynArts, CalcHep/CompHep and also MicrOmegas, streamlining the implementation process. To address Mathematica's limitations with heavy computations, SARAH produces source code for SPheno, a Fortran tool for efficient numerical evaluations.

2.2.1 Installation of SARAH

Here's a step-by-step guide for installing SARAH on a Windows system:-

1. **Install Mathematica:**
   Ensure that there is a compatible version of Mathematica installed on the Windows system. If not, then one can install it from the Wolfram's Website.

2. **Download SARAH:**
   Visit the official SARAH website or the GitHub repository to download the latest version of SARAH. The package is typically provided as a compressed file (e.g., ZIP or TAR).

3. **Extract the SARAH Package:**
   Locate the downloaded compressed file, right-click on it, and select "Extract All..." to unpack it. Choose a suitable directory for extraction

4. **Set Up the SARAH Path in Mathematica:**
   Open Mathematica and create a new notebook. Set the SARAH path by executing the following command, replacing C:\SARAH with the path where you extracted SARAH:

   ⟨⟨"C:\\SARAH-X.Y.Z\\SARAH.m"

   Here, X.Y.Z have to replaced by version of SARAH. Every time we start mathematica we have to execute this command to load SARAH in Mathematica, otherwise it won't work.

5. **Starting a Model:**
   After successful setup of SARAH in Mathematica we can start any model and get analytical information about that model using various commands. In order to initalize a model we have to use the following command:-

   Start[ModelName];

2.2.2 Defining a Model in SARAH

SARAH includes a variety of pre-implemented models like SM and MSSM etc., which we can implement. But along with these we can also define new models in SARAH and then get analytical information from them.

To define a new model in SARAH, navigate to the directory C:\SARAH (or any other PATH where SARAH is located), where each folder holds the files corresponding to a specific model. For example, the SM folder contains files for the Standard Model.Each model directory comprises four essential files:

- model.m: This file outlines the fundamental definitions of the model (replace "model" with the specific model name).
- parameters.m: Contains additional information about the model's parameters.
- particles.m: Focuses on the particles in the model, providing details not included in model.m.
- SPheno.m: This file is necessary only if you intend to create a SPheno module for the model.

Now we will create a new model named Scotogenic Model, which is simpler that SM model. The Steps for defining the model are-

- Begin by creating a new folder named Scotogenic in PATH\SARAH-X.Y.Z\Models.
- After this we have to create the four files mentioned above- Scotogenic.m, parameters.m, particles.m and SPheno.m.
- . After successfully creating these files and adding all the details we are ready to implement the model to get different informations

The Images of model files including model.m, parameters.m, particles.m, and SPheno.m, are provided in Appendix A for reference.

2.2.3 Exploring Scotogenic Model

Once we have successfully made all the model files our model is ready for implementation. We can Start the Scotogenic model using the command-

Start[Scotogenic];

Once the Model is successfully loaded in SARAH we can execute different kinds of commands to get analytical information about the model.
Some of the commands that we will execute are:-

- **Tadpole Equations:**

  Tadpole equations are mathematical conditions used in particle physics models to ensure that the potential energy of the fields is stable. In SARAH, once you define a model, it automatically calculates these equations for you. They help identify the values of fields (like scalar fields) at which the potential is minimized.
  We can calculate these using the command-

  TadpoleEquation [v]

  This will give us the tadpole equation of the scalar field.

- **Masses:**

  In SARAH, you can calculate various masses of particles in a model. Once you define your model, SARAH generates mass matrices, which show how particles interact and acquire mass. There are two ways of doing this depending on whether the mass eigenstate we are interested in is a mixture of gauge eigenstates or not.
  If it is we use the commands-

  MassMatrix [state]

where state must be replaced by the name of the mass eigenstate. For example, [Fe], [Chi]

Now, when the mass of a state does not mix with other fields we must use the command-

Mass [state] /. Masses [EWSB]

where, again, state must be replaced by the name of the specific mass eigenstate. As a prime example we can take [hh]

- **Vertices:**

In SARAH, vertices refer to the points where particles interact with each other in a model. Using SARAH, we can automatically calculate all the interaction vertices for a given particle physics model. This includes details about which particles interact and how strong the interactions are. These vertices are essential for studying particle interactions and are used in various simulations and calculations. The commands which we can use are-

Vertex [state 1 , state 2 , state 3]

or,

Vertex [state 1 , state 2 , state 3 , state 4]

depending on the number of particles involved in the vertex. Here, state 1, state 2,state 3 and state 4 are mass eigenstates. For example, Vertex[bar [Fe] , Fe , hh]. The result of this command is an array that includes all possible Lorentz structures appearing in the interaction vertex and the corresponding coefficient.

- **Renormalization Group Equations (RGEs):**

Renormalization Group Equations (RGEs) describe how the parameters of a particle physics model, like masses and coupling constants, change with energy. In SARAH, you can automatically calculate these RGEs for models. This helps in understanding how the behavior of particles and interactions evolves at different energy scales, which is important for studying theories beyond the Standard Model.

SARAH generates these equations up to two-loop level, allowing detailed analysis of how a model behaves at high energies. But we will generate only One loop because for non-supersymmetric model both level require a lot of time to get execute.

The command for this is-

CalcRGEs [TwoLoop -⟩ False]

The analytical results for the RGEs are saved in several arrays.

2.2.4 LaTeX Format Output

Finally, we can export all the analytical results derived by SARAH into LaTeX format. After compiling, this produces a PDF that contains all the model's information in an organized, readable form. The LaTeX output from SARAH is highly convenient—not only is it easy to read, but we can also directly copy the LaTeX code for use in our own publications, saving time and avoiding the tedious task of manually typing complex formulas. To generate the LaTeX output for our model, we simply use the following commands.

ModelOutput [EWSB]

MakeTeX []

The first command instructs SARAH to perform various computations, with the results saved in multiple directories under PATH\SARAH-X.Y.Z\Output\Scotogenic\EWSB The results of these commands are put in the directory PATH\SARAH-X.Y.Z\Output\Scotogenic\EWSB\TeX

In order to generate the pdf file with all the information, we just have to go to that directory and open the TeX file with any LaTeX editor or online editor Overleaf.

3 Results

The results of this project are compiled and presented in the accompanying Mathematica notebook. This notebook contains all the calculations, analyses, and outputs generated using SARAH. It includes the model definitions, parameter setups, RGEs, and other analytical expressions derived throughout the project. Additionally, the notebook showcases the use of SARAH to automate key tasks, such as exporting LaTeX files for further documentation.

SARAH: *Mathematica* Package

SriVipra'24 Project

Setting up SARAH in *Mathematica*

```
<<
 "C:\\Users\\LENOVO\\Documents\\SriVipra\\SARAH-4.15.3\\SARAH-4.15.3\\SARAH.m"
```

SARAH 4.15.3

by Florian Staub, Mark Goodsell and Werner Porod, 2020

contributions by M. Gabelmann, K. Nickel

References:

Comput.Phys.Commun.181 (2010) 1077-1086. (arXiv:0909.2863)

Comput.Phys.Commun.182 (2011) 808-833. (arXiv:1002.0840)

Comput.Phys.Commun.184 (2013) 1792-1809. (arXiv:1207.0906)

Comput.Phys.Commun.185 (2014) 1773-1790. (arXiv:1309.7223)

Eur.Phys.J.C 74 (2014) 8, 2992 (arXiv:1405.1434)

Eur.Phys.J.C 75 (2015) 1, 32 (arXiv:1411.0675)

Eur.Phys.J.C 75 (2015) 6, 290 (arXiv:1503.03098)

Eur.Phys.J.C 77 (2017) 11, 758 (arXiv:1703.09237

Eur.Phys.J.C 77 (2017) 11, 757 (arXiv:1706.05372)

Eur.Phys.J.C 78 (2018) 8, 649 (arXiv:1805.07306)

Download and Documentation:

http://sarah.hepforge.org

Start evaluation of a model with:

Start["Name of Model"]

e.g. Start["MSSM"] or Start["NMSSM","CKM"]

To get a list with all installed models, use ShowModels

The Setup of SARAH in *Mathematica* is completed. Now we can load any SARAH model according to the requirement.

Now to implement any Model, we have to use the folloing code:

```
Start["MSSM"]
```

Once the setup of the Model is completed we can use it to calculate different parameters.

Defining a Model in SARAH

Along with Models already present in SARAH we can also add new models. We will implement Scotogenic model in SARAH. For this we must create a new folder called Scotogenic in Models in SARAH.

In this folder we have to make four files- Scotogenic.m, parameters.m, particles.m and SPheno.m. After successfully creating these files and adding all the details we are ready to implement the model to get different informations.

```
(* First we will setup the Model *)
Start["Scotogenic"]
```

```
Preparing arrays

 ... checking Directory:
 C:\Users\LENOVO\Documents\SriVipra\SARAH-4.15.3\SARAH-4.15.3\Models\
```

Model files loaded

```
  Model     : Scotogenic

  Author(s): Mahi Garg

  Date      : 2024-09-10

*******************************************************

Loading Susyno functions for the handling of Lie Groups

Based on Susyno v.2.0 by Renato Fonseca (1106.5016)

webpage: web.ist.utl.pt/renato.fonseca/susyno.html

*******************************************************
```

Initialization

```
Checking model files: SARAH`DynamicCheckModelFiles

Initialize gauge groups: SARAH`DynamicInitGaugeG

Initialize field: SARAH`DynamicInitFields

Preprocessing necessary information: SARAH`DynamicInitMisc

Checking for anomalies: SARAH`DynamicCheckAnomalies
```

Derive Lagrangian

```
Calculate kinetic Terms

  ... for scalars: SARAH`DynamicKineticScalarNr/8 (SARAH`DynamicKineticScalarName)

  ... for fermions: SARAH`DynamicKineticFermionNr/8 (SARAH`DynamicKineticFermionName)

Calculate vector self-interactions: SARAH`DynamicVectorNr/3 (SARAH`DynamicVectorName)

Calculate gauge transformations: SARAH`DynamicGaugeTNr/11 (SARAH`DynamicGaugeTName)

Eleminate sums

Calc Complete Lagrangian
```

Evolve States: GaugeES

```
Adding terms to the Lagrangian:

 ... adding: Yn Et.n.l + Yu H.u.q + Yd conj[H].d.q + Ye conj[H].e.l (
 SARAH`DynamicStatusAddTerms[Yn Et.Susyno`LieGroups`n.SARAH`l +
   SARAH`Yu H.u.SARAH`q + SARAH`Yd Susyno`LieGroups`conj[H].SARAH`d.SARAH`q +
   SARAH`Ye Susyno`LieGroups`conj[H].Susyno`LieGroups`e.SARAH`l] )

 ... adding: Mn n.n/2 (
 SARAH`DynamicStatusAddTerms[Mn Susyno`LieGroups`n.Susyno`LieGroups`n/2])

 ... adding: mH2 conj[H].H - 1/2 lambda1 conj[H].H.conj[H].H (
 SARAH`DynamicStatusAddTerms[mH2 Susyno`LieGroups`conj[H].H -
   1/2 lambda1 Susyno`LieGroups`conj[H].H.Susyno`LieGroups`conj[H].H] )

 ... adding: -mEt2 conj[Et].Et - 1/2 lambda2 conj[Et].Et.conj[Et].Et (
 SARAH`DynamicStatusAddTerms[-mEt2 Susyno`LieGroups`conj[Et].Et -
   1/2 lambda2 Susyno`LieGroups`conj[Et].Et.Susyno`LieGroups`conj[Et].Et] )

 ... adding: -lambda4 conj[H].Et.conj[Et].H - lambda3 conj[H].H.conj[Et].Et
  (SARAH`DynamicStatusAddTerms[
    -lambda4 Susyno`LieGroups`conj[H].Et.Susyno`LieGroups`conj[Et].H -
     lambda3 Susyno`LieGroups`conj[H].H.Susyno`LieGroups`conj[Et].Et] )

 ... adding: -1/2 lambda5 conj[H].Et.conj[H].Et (
 SARAH`DynamicStatusAddTerms[
  -1/2 lambda5 Susyno`LieGroups`conj[H].Et.Susyno`LieGroups`conj[H].Et] )
Rotate Lagrangian: SARAH`DynamicRotateLag[1]/14

Derive ghost terms:

  ... generate gauge fixing terms:
 SARAH`DynamicGFnr[SARAH`GaugeES]/3 (SARAH`DynamicGFname[SARAH`GaugeES])

  ... calculate Ghost interactions

Calc Mixings of Matter Fields

Save information (SARAH`DynamicSaveInfo[SARAH`GaugeES])
```

Evolve States: EWSB

```
Parametrize Higgs Sector

   Update gauge transformations: SARAH`DynamicUGT[SARAH`UGTvev[2]]
 /21 (SARAH`DynamicUGTname[SARAH`UGTvev[2]])

Calc mass matrices gauge sector:
 SARAH`DynamicMMgaugeNr[SARAH`EWSB]/2(SARAH`DynamicMMgaugeName[SARAH`EWSB])

   Update gauge transformations: SARAH`DynamicUGT[SARAH`UGTgaugeMM[2]]
 /28 (SARAH`DynamicUGTname[SARAH`UGTgaugeMM[2]])

Rotate Lagrangian: SARAH`DynamicRotateLag[2]/14
```

```
Derive ghost terms:

  ... generate gauge fixing terms:
 SARAH`DynamicGFnr[SARAH`EWSB]/4 (SARAH`DynamicGFname[SARAH`EWSB])

  ... calculate Ghost interactions

Calc Mixings of Matter Fields

   Calculate mass matrices

 SARAH`DynamicNrMass[{{{Susyno`LieGroups`conj[nR]}, {Susyno`LieGroups`conj[nR]}},/5 (
   {{vL}, {vL}}, {{dL}, {Susyno`LieGroups`conj[dR]}},
   {{uL}, {Susyno`LieGroups`conj[uR]}}, {{eL}, {Susyno`LieGroups`conj[eR]}}}]

 SARAH`DynamicNameMass[{{{Susyno`LieGroups`conj[nR]}, {Susyno`LieGroups`conj[nR]}},)
   {{vL}, {vL}}, {{dL}, {Susyno`LieGroups`conj[dR]}},
   {{uL}, {Susyno`LieGroups`conj[uR]}}, {{eL}, {Susyno`LieGroups`conj[eR]}}}]

   Update gauge transformations: SARAH`DynamicUGT[SARAH`UGTmatterMM[2]]
 /32 (SARAH`DynamicUGTname[SARAH`UGTmatterMM[2]])

Calculate Tadpole Equations

Save information (SARAH`DynamicSaveInfo[SARAH`EWSB])
```

Finishing

```
Calculate final Lagrangian

Cleaning up

   ... add matrix products

   ... checking for massless particles

   ... calculating tree level masses (SARAH`DynamicCalcTreeMasses)

   ... simplify mass matrices

Numerical calculations (if necessary)

   ... checking for spectrum file: SARAH`DynamicSpectrumFileInput

   ... reading parameter values and dependences

   ... calculate mixing matrices

Checking for CP even and odd scalars


All Done. Scotogenic is ready!

(Model initialized in 1.328s)


Are you unfamiliar with SARAH? Use SARAH`FirstSteps
```

Scotogenic Model is now setup successfully. Now we will use different commands to get analytical information about the model.
So, let us first obtain the Tadpole Equations. The command used is-

TadpoleEquation[v]

$$-\text{mH2}\ v + \frac{\text{lambda1}\ v^3}{2} == 0$$

This can be further minimized using the command-

```
Solve[TadpoleEquation[v], mH2]
```

$$\left\{\left\{mH2 \to \frac{lambda1\ v^2}{2}\right\}\right\}$$

We can obtain the complete list of tadpole equations of a model using-

```
TadpoleEquations[EWSB]
```

$$\left\{-mH2\ v + \frac{lambda1\ v^3}{2},\ 0\right\}$$

Now we will print some masses. Now there are two ways for doing this epending on whether the mass eigenstate we are interested in is a mixture of gauge eigenstates or not.
When it is, we must print the mass matrix of the
complete set of mass eigenstates. This is done with:

```
(* Printing Mass Matrix of Singlet Fermions*)
MassMatrix[Chi]
```

$$\left\{\left\{-Mn[1,1],\ -\frac{1}{2}Mn[1,2]-\frac{1}{2}Mn[2,1],\ -\frac{1}{2}Mn[1,3]-\frac{1}{2}Mn[3,1]\right\},\right.$$
$$\left\{-\frac{1}{2}Mn[1,2]-\frac{1}{2}Mn[2,1],\ -Mn[2,2],\ -\frac{1}{2}Mn[2,3]-\frac{1}{2}Mn[3,2]\right\},$$
$$\left.\left\{-\frac{1}{2}Mn[1,3]-\frac{1}{2}Mn[3,1],\ -\frac{1}{2}Mn[2,3]-\frac{1}{2}Mn[3,2],\ -Mn[3,3]\right\}\right\}$$

If required, we can simplify the expression with the command-

```
MassMatrix[Chi] /. Mn[i_, j_] :> If[i > j, Mn[j, i], Mn[i, j]]
```

```
{{-Mn[1, 1], -Mn[1, 2], -Mn[1, 3]},
 {-Mn[1, 2], -Mn[2, 2], -Mn[2, 3]}, {-Mn[1, 3], -Mn[2, 3], -Mn[3, 3]}}
```

Now, when the mass of a state does not mix with other fields we must use the command-

```
(* Printing mass of Higgs Boson*)
Mass[hh] /. Masses[EWSB]
```

$$-mH2 + \frac{3\ lambda1\ v^2}{2}$$

This is not the exact mass of Higgs Boson but this is the squared mass of higgs boson, This is because minimization condition has not been been applied yet. So, applying the tadpole equation-

```
solTadpole = Solve[TadpoleEquation[v], mH2];
Mass[hh] /. Masses[EWSB] /. solTadpole
```

$$\{lambda1\ v^2\}$$

This gives us the final expression for mass of Higgs Boson

We can also obtain intersection vertices using this model. The command for the same is-

```
Vertex[{bar[Fe], Fe, hh}]
```

```
{{bar[Fe[{gt1}]], Fe[{gt2}], hh},
 {1/Sqrt[2] i sum[j2, 1, 3, conj[Ve[gt2, j2]] sum[j1, 1, 3, conj[Ue[gt1, j1]] Ye[j1, j2]]],
  PL},
 {1/Sqrt[2] i sum[j2, 1, 3, sum[j1, 1, 3, conj[Ye[j1, j2]] Ue[gt2, j1]] Ve[gt1, j2]], PR}}
```

SARAH also identifies when the vertex does not exist and return zero.

```
Vertex[{Fv, Chi, hh}]
```

```
{{Fv[{gt1}], Chi[{gt2}], hh}, {0, PL}, {0, PR}}
```

We can also obtain the renormalization group equations (RGEs) for all the parameters of the model using-

```
CalcRGEs[TwoLoop → False]
```

Calculate non−supersymmetric RGEs

```
Initializing Invariants...
```

Fermion bilinear

```
                              (Time needed so far: 1.719s)
```

Fermion chains

```
                              (Time needed so far: 11.157s)
```

Scalar bilinear

```
      ... 1-loop
```

Scalar quartic

```
      ... 1-loop
```

Scalar cubic

```
      ... 1-loop
      ... 2-loop
                              (Time needed so far: 24.391s)
All necessary invariants are ready (Time needed: 24.391s)


Calculation of beta functions...
```

Calculate Beta Functions for Gauge Couplings

```
   Calculating SARAH`DynamicProgressRGE[SARAH`GAUGE]
 /3.(SARAH`DynamicCoupProgess[SARAH`GAUGE])
```

Calculate anomalous dimensions of scalars

```
   Calculating SARAH`DynamicProgressRGE[SARAH`GSIJ]
 /4.(SARAH`DynamicCoupProgess[SARAH`GSIJ])
```

Calculate hat(gamma) for running VEVs

```
   Calculating SARAH`DynamicProgressRGE[SARAH`GSijHat]
 /4.(SARAH`DynamicCoupProgess[SARAH`GSijHat])
```

Calculate anomalous dimensions of fermions

```
Calculating SARAH`DynamicProgressRGE[SARAH`GFIJ]
/6.(SARAH`DynamicCoupProgess[SARAH`GFIJ])
```

Calculate Beta Functions for Yukawa–like interactions

```
Calculating SARAH`DynamicProgressRGE[SARAH`YIJK]
/4.(SARAH`DynamicCoupProgess[SARAH`YIJK])
```

Calculate Beta Functions for bilinear Fermion interactions

```
Calculating SARAH`DynamicProgressRGE[SARAH`MFIJ]
/1.(SARAH`DynamicCoupProgess[SARAH`MFIJ])
```

Calculate Beta Functions for scalar 4–point interactions

```
Calculating SARAH`DynamicProgressRGE[SARAH`LIJKL]
/5.(SARAH`DynamicCoupProgess[SARAH`LIJKL])
```

Calculate Beta Functions for scalar 3–point interactions

```
Nothing to do.
```

Calculate Beta Functions for bilinear scalar interactions

```
Calculating SARAH`DynamicProgressRGE[SARAH`MSIJ]
/2.(SARAH`DynamicCoupProgess[SARAH`MSIJ])
```

Calculate Beta Functions for VEVs

```
Calculating SARAH`DynamicProgressRGE[SARAH`VEVI]
/1.(SARAH`DynamicCoupProgess[SARAH`VEVI])
Writing Mathematica code to evaluate RGEs


Finished with the calculation of the RGEs. Time needed: 25.703s
```

The results are saved in C:\Users\LENOVO\Documents\SriVipra\SARAH–4.15.3\SARAH–4.15.3\Output\Scotogenic\RGEs\

LaTeX Format Output

So, this is some of the analytical information we can obtain using SARAH.
We can also export this information LaTeX format so that we obtain a pdf file which is easy to read and use.
We can generate LaTeX output using the following command-

```
ModelOutput[EWSB]
MakeTeX[]
```

```
Checking model for missing definitions

Generate Directories

Building Particle List
```

Calculate all vertices

Three Scalar – Interactions

```
Found 14 potential vertices. Calculating
SARAH`progressNrGV[SARAH`SSS]/14 . (SARAH`progressCurrentGV[SARAH`SSS])
```

Four Scalar – Interactions

```
Found 36 potential vertices. Calculating
SARAH`progressNrGV[SARAH`SSSS]/36 . (SARAH`progressCurrentGV[SARAH`SSSS])
```

Two Scalar – One Vector Boson – Interactions

Found 64 potential vertices. Calculating
SARAH`progressNrGV[SARAH`SSV]/64 . (SARAH`progressCurrentGV[SARAH`SSV])

One Scalar – Two Vector Boson – Interactions

Found 27 potential vertices. Calculating
SARAH`progressNrGV[SARAH`SVV]/27 . (SARAH`progressCurrentGV[SARAH`SVV])

Two Scalar – Two Vector Boson – Interactions

Found 140 potential vertices. Calculating
SARAH`progressNrGV[SARAH`SSVV]/140 . (SARAH`progressCurrentGV[SARAH`SSVV])

Three Vector Boson – Interactions

Found 21 potential vertices. Calculating
SARAH`progressNrGV[SARAH`VVV]/21 . (SARAH`progressCurrentGV[SARAH`VVV])

Two Fermion – One Scalar – Interactions

Found 14 potential vertices. Calculating
SARAH`progressNrGV[SARAH`FFS]/14 . (SARAH`progressCurrentGV[SARAH`FFS])

Two Fermion – One Vector Boson – Interactions

Found 34 potential vertices. Calculating
SARAH`progressNrGV[SARAH`FFV]/34 . (SARAH`progressCurrentGV[SARAH`FFV])

Four Vector Boson – Interactions

Found 36 potential vertices. Calculating
SARAH`progressNrGV[SARAH`VVVV]/36 . (SARAH`progressCurrentGV[SARAH`VVVV])

Two Ghost – One Vector Boson – Interactions

Found 65 potential vertices. Calculating
SARAH`progressNrGV[SARAH`GGV]/65 . (SARAH`progressCurrentGV[SARAH`GGV])

Two Ghost – One Scalar – Interactions

Found 48 potential vertices. Calculating
SARAH`progressNrGV[SARAH`GGS]/48 . (SARAH`progressCurrentGV[SARAH`GGS])

Two Scalar – One Auxiliary – Interactions

Simplify Vertices

Writing vertices to files

All vertices calculated. (Time needed: 25.297s)

The vertices are saved in
C:\Users\LENOVO\Documents\SriVipra\SARAH-4.15.3\SARAH-4.15.3\Output\Scotogenic\EWSB\Vertices\

Generate LaTeX files

Writing Fields and Lagrangian to TeX-File

Writing Particle Content to TeX-File

Write VEVs to TeX-File

Write Flavor Decomposition to TeX-File

Writing Mass Matrices to TeX-File

Writing Tadpole Equations to TeX-File

Writing RGEs to TeX-File

TeXOutput::NoLoops :
 Loop corrections not calculated so far. Skipping this parts.

Use CalcLoopCorrections[States] to calculated RGEs and start MakeTeX again to include them in the output.

```
Write Clebsch-Gordan Coefficients

Writing Vertices to TeX-File


Done. Output is in
 C:\Users\LENOVO\Documents\SriVipra\SARAH-4.15.3\SARAH-4.15.3\Output\Scotogenic\EWSB\
   TeX\

Use Script MakePDF.sh (Linux) or MakePDF.bat (Windows) to generate pdf file.
```

4 Conclusion

In this project, we successfully explored the use of the SARAH package for analyzing particle physics models beyond the Standard Model. We gained hands-on experience with defining new models, generating analytical expressions, and exporting results for further use in other computational tools like MicrOMEGAs. Through this process, we also developed a foundational understanding of the theoretical concepts, such as Tadpole Equations, Masses, vertices , Renormalization Group Equations (RGEs) and dark matter interactions. The skills and knowledge acquired during this project will prove valuable for future research endeavors in particle physics and cosmology.

5 Appendix A: Model Files for Scotogenic Model

First we have the details of Scotogenic.m file. After that we have parameter.m file followed by particle.m and then SPheno.m.

```
Off[General::spell]

Model`Name = "Scotogenic";
Model`NameLaTeX ="Scotogenic Model";
Model`Authors = "Mahi Garg";
Model`Date = "2024-09-10";
```

```
(*------------------------------------------*)
(*   Particle Content*)
(*------------------------------------------*)
(* Global Symmetries *)
Global [[1]] = {Z[2] , Z2};

(* Gauge Groups *)

Gauge[[1]]={B,   U[1], hypercharge, g1, False,  1};
Gauge[[2]]={WB, SU[2], left,        g2, True,   1};
Gauge[[3]]={G,  SU[3], color,       g3, False,  1};
```

```
(* Matter Fields *)

FermionFields[[1]] = {q, 3, {uL, dL},     1/6, 2,  3, 1};
FermionFields[[2]] = {l, 3, {vL, eL},    -1/2, 2,  1, 1};
FermionFields[[3]] = {d, 3, conj[dR],     1/3, 1, -3, 1};
FermionFields[[4]] = {u, 3, conj[uR],    -2/3, 1, -3, 1};
FermionFields[[5]] = {e, 3, conj[eR],       1, 1,  1, 1};
FermionFields[[6]] = {n, 3, conj[nR],       0, 1,  1, -1};

ScalarFields[[1]] =  {H,  1, {Hp, H0},     1/2, 2,  1,  1};
ScalarFields[[2]] =  {Et, 1, {etp,et0},    1/2, 2,  1, -1};
```

```
(*---------------------------------------------*)
(*   DEFINITION                                *)
(*---------------------------------------------*)

NameOfStates={GaugeES, EWSB};

(* ----- Before EWSB ----- *)

DEFINITION[GaugeES][LagrangianInput]=
{
    {LagFer, {AddHC -> True}},
    {LagNV,{AddHC -> True}},
    {LagH, {AddHC -> False}},
    {LagEt,{AddHC -> False}},
    {LagHEt, {AddHC -> False}},
    {LagHEtHC,{AddHC -> True}}
};
```

```
LagFer = Yd conj[H].d.q + Ye conj[H].e.l + Yu H.u.q + Yn Et.n.l;
LagNV = Mn/2 n.n;
LagH = -(- mH2 conj[H].H + 1/2 lambda1 conj[H].H.conj[H].H);
LagEt = -(+ mEt2 conj[Et].Et + 1/2 lambda2 conj[Et].Et.conj[Et].Et);
LagHEt = -(+ lambda3 conj[H].H.conj[Et].Et + lambda4 conj[H].Et.conj[Et].H);
LagHEtHC = -(+ 1/2 lambda5 conj[H].Et.conj[H].Et);
```

```
(* Gauge Sector *)

DEFINITION[EWSB][GaugeSector] =
{
  {{VB,VWB[3]},{VP,VZ},ZZ},
  {{VWB[1],VWB[2]},{VWp,conj[VWp]},ZW}
};




(* ----- VEVs ---- *)

DEFINITION[EWSB][VEVs]=
{    {H0, {v, 1/Sqrt[2]}, {Ah, I/Sqrt[2]}, {hh, 1/Sqrt[2]}},
     {et0, {0, 0}, {etI, I/Sqrt[2]}, {etR, 1/Sqrt[2]}}
};


DEFINITION[EWSB][MatterSector]=
{
     {{conj[nR]}, {X0, ZX}},
     {{vL},  {VL, Vv}},
     {{{dL},  {conj[dR]}},  {{DL, Vd}, {DR, Ud}}},
     {{{uL}, {conj[uR]}}, {{UL,Vu}, {UR,Uu}}},
     {{{eL}, {conj[eR]}}, {{EL,Ve}, {ER,Ue}}}
};
```

```
(*-----------------------------------------------------*)
(* Dirac-Spinors *)
(*-----------------------------------------------------*)

DEFINITION[EWSB][DiracSpinors]=
{
 Fd -> {  DL, conj[DR]},
 Fe -> {  EL, conj[ER]},
 Fu -> {  UL, conj[UR]},
 Fv -> {  VL, conj[VL]},
 Chi -> {  X0, conj[X0]}
};

DEFINITION[EWSB][GaugeES]={
 Fd1 -> {  FdL, 0},
 Fd2 -> {  0, FdR},
 Fu1 -> {  Fu1, 0},
 Fu2 -> {  0, Fu2},
 Fe1 -> {  Fe1, 0},
 Fe2 -> {  0, Fe2}};
```

```
ParameterDefinitions = {

{g1,        { Description ->  "Hypercharge-Coupling"}},
{g2,        { Description ->  "Left-Coupling"}},
{g3,        { Description ->  "Strong-Coupling"}},

{AlphaS,    {Description ->  "Alpha Strong"}},
{e,         { Description ->  "electric charge"}},

{Gf,        { Description ->  "Fermi's constant"}},
{aEWinv,    { Description ->  "inverse weak coupling constant at mZ"}},

{Yu,        { Description ->  "Up-Yukawa-Coupling",
             DependenceNum ->   Sqrt[2]/v* {{Mass[Fu,1],0,0},
                                            {0, Mass[Fu,2],0},
                                            {0, 0, Mass[Fu,3]}}}},

{Yd,        { Description ->  "Down-Yukawa-Coupling",
              DependenceNum ->   Sqrt[2]/v* {{Mass[Fd,1],0,0},
                                            {0, Mass[Fd,2],0},
                                            {0, 0, Mass[Fd,3]}}}},

{Ye,        { Description ->  "Lepton-Yukawa-Coupling",
              DependenceNum ->   Sqrt[2]/v* {{Mass[Fe,1],0,0},
                                            {0, Mass[Fe,2],0},
                                            {0, 0, Mass[Fe,3]}}}},


{ThetaW,    { Description ->  "Weinberg-Angle",
              DependenceNum ->  ArcSin[Sqrt[1 - Mass[VWp]^2/Mass[VZ]^2]]  }},

{ZZ, {Description ->  "Photon-Z Mixing Matrix"}},
{ZW, {Description ->  "W Mixing Matrix",
       Dependence ->    1/Sqrt[2] {{1, 1},
                  {I,-I}} }},


{Vu,        {Description -> "Left-Up-Mixing-Matrix"}},
{Vd,        {Description -> "Left-Down-Mixing-Matrix"}},
{Uu,        {Description -> "Right-Up-Mixing-Matrix"}},
{Ud,        {Description -> "Right-Down-Mixing-Matrix"}},
{Ve,        {Description -> "Left-Lepton-Mixing-Matrix"}},
{Ue,        {Description -> "Right-Lepton-Mixing-Matrix"}},

(* Scalar sector*)

{v,          { Description ->  "EW-VEV",
               DependenceNum ->  Sqrt[4*Mass[VWp]^2/(g2^2)],
               DependenceSPheno ->  None,
               OutputName ->  vvSM}},

{mH2,        { Description ->  "SM Higgs Mass Parameter"}},
{mEt2,       {LaTeX ->  "m_\\eta^2",
              LesHouches ->  {HDM,1},
              OutputName ->  mEt2}},

{lambda1,    {LaTeX ->  "\\lambda_1",
              LesHouches ->  {HDM, 2}},
              OutputName ->  lam1 }},
```

```
{lambda2,     {LaTeX ->  "\\lambda_2",
               LesHouches ->  {HDM, 3},
               OutputName ->  lam2 }},

{lambda3,     {LaTeX ->  "\\lambda_3",
               LesHouches ->  {HDM, 4},
               OutputName ->  lam3 }},
{lambda4,     {LaTeX ->  "\\lambda_4",
               LesHouches ->  {HDM, 5},
               OutputName ->  lam4 }},
{lambda5,     {Real ->  True,
               LaTeX ->  "\\lambda_5",
               LesHouches ->  {HDM, 6},
               OutputName ->  lam5 }},

(* Fermion sector*)
{Yn,          {LaTeX ->   "Y_N",
               LesHouches ->  YN,
               OutputName ->  Yn}},
{Mn,          {LaTeX ->   "M_N",
               LesHouches ->  MN,
               OutputName ->  Mn}},
{ZX,          {LaTeX ->   "Z^{\\chi^0}",
               LesHouches ->  ZXMIX,
               OutputName ->  ZX}},
{Vv,          {Description ->  "Neutrino-Mixing-Matrix"}}
 };
```

```
ParticleDefinitions[GaugeES] = {
      {H0,  {     PDG ->   {0},
                 Width ->   0,
                 Mass ->   Automatic,
                 FeynArtsNr ->   1,
                 LaTeX ->   "H^0",
                 OutputName ->   "H0" }},


      {Hp,  {     PDG ->   {0},
                 Width ->   0,
                 Mass ->   Automatic,
                 FeynArtsNr ->   2,
                 LaTeX ->   "H^+",
                 OutputName ->   "Hp" }},

      {et0, {     PDG ->   {0},
                 Width ->   0,
                 Mass ->    Automatic,
                 LaTeX ->   "\\eta^0",
                 OutputName ->   "et0"}},

      {etp, {     PDG ->   {0},
                 Width ->   0,
                 Mass ->    Automatic,
                 LaTeX ->   "\\eta^+",
                 OutputName ->   "etp"}},

      {VB,   { Description ->   "B-Boson"}},
      {VG,   { Description ->   "Gluon"}},
      {VWB,  { Description ->   "W-Bosons"}},
      {gB,   { Description ->   "B-Boson Ghost"}},
      {gG,   { Description ->   "Gluon Ghost" }},
      {gWB,  { Description ->   "W-Boson Ghost"}}

      };




  ParticleDefinitions[EWSB] = {


    {hh   ,  {  Description ->   "Higgs",
                 PDG ->   {25},
                 PDG.IX ->   {101000001},
                 Mass ->    Automatic}},
     {Ah   ,  {  Description ->   "Pseudo-Scalar Higgs",
                 PDG ->   {0},
                 PDG.IX ->  {0},
                 Mass ->   {0},
                 Width ->   {0} }},


     {Hp,     { Description ->   "Charged Higgs",
                 PDG ->   {0},
                 PDG.IX ->  {0},
                 Width ->   {0},
                 Mass ->    {0},
```

```
                  LaTeX ->   {"H^+","H^-"},
                  OutputName ->   {"Hp","Hm"},
                  ElectricCharge ->  1
                  }},

       {etR,   { Description ->   "CP-even eta scalar",
                  PDG ->   {1001},
                  Mass ->   LesHouches,
                  ElectricCharge ->   0,
                  LaTeX ->   "\\eta_R",
                  OutputName ->   "etR"
                  }},

       {etI,     {Description ->   "CP-odd eta scalar",
                  PDG ->   {1002},
                  Mass ->   LesHouches,
                  ElectricCharge ->   0,
                  LaTeX ->   "\\eta_I",
                  OutputName ->   "etI"
                  }},

       {etp,     {Description ->   "Charged eta scalar",
                  PDG ->   {1003},
                  Mass ->   LesHouches,
                  ElectricCharge ->   1,
                  LaTeX ->   "\\eta^+",
                  OutputName ->   "etp"
                  }},

       {VP,   { Description ->   "Photon"}},
       {VZ,   { Description ->   "Z-Boson",
                    Goldstone ->   Ah }},
       {VG,   { Description ->   "Gluon" }},
       {VWp,  { Description ->   "W+ - Boson",
                   Goldstone ->   Hp }},
       {gP,   { Description ->   "Photon Ghost"}},
       {gWp,  { Description ->   "Positive W+ - Boson Ghost"}},
       {gWpC, { Description ->   "Negative W+ - Boson Ghost" }},
       {gZ,   { Description ->   "Z-Boson Ghost" }},
       {gG,   { Description ->   "Gluon Ghost" }},


       {Fd,   { Description ->   "Down-Quarks"}},
       {Fu,   { Description ->   "Up-Quarks"}},
       {Fe,   { Description ->   "Leptons" }},
       {Fv,   { Description ->   "Neutrinos" }},
       {Chi,  { Description ->     "Singlet Fermions",
                PDG ->   {1012,1014,1016},
                Mass ->   LesHouches,
                ElectricCharge ->   0,
                LaTeX ->   "N",
                OutputName ->   "N"  }}
};



 WeylFermionAndIndermediate = {

   {H , {   PDG -> {0},
            Width -> 0,
```

```
         Mass -> Automatic,
         LaTeX -> "H",
         OutputName -> "" }},
{Et, {LaTeX ->   " \\ eta " }},
{dR, {LaTeX ->   "d_R" }},
{eR, {LaTeX ->   "e_R" }},
{lep,{LaTeX ->   "l" }},
{uR, {LaTeX ->   "u_R" }},
{q,  {LaTeX ->   "q" }},
{eL, {LaTeX ->   "e_L" }},
{dL, {LaTeX ->   "d_L" }},
{uL, {LaTeX ->   "u_L" }},
{vL, {LaTeX ->   "\\nu_L" }},

{DR, {LaTeX ->   "D_R" }},
{ER, {LaTeX ->   "E_R" }},
{UR, {LaTeX ->   "U_R" }},
{EL, {LaTeX ->   "E_L" }},
{DL, {LaTeX ->   "D_L" }},
{UL, {LaTeX ->   "U_L" }},
{X0 ,{LaTeX ->   " X^0 " }},
{VL ,{LaTeX ->   " V_L " }} ,
{n , {LaTeX ->   " N " }},
{nR ,{LaTeX ->   " \\ nu_R " }}
     };
```

```
OnlyLowEnergySPheno = True;

MINPAR={
  {1,Lambda1Input},
  {2,Lambda2Input},
  {3,Lambda3Input},
  {4,Lambda4Input},
  {5,Lambda5Input},
  {6,mEt2Input}
};
```

```
ParametersToSolveTadpoles = {mH2};

BoundaryLowScaleInput={
 {lambda1,Lambda1Input},
 {lambda2,Lambda2Input},
 {lambda3,Lambda3Input},
 {lambda4,Lambda4Input},
 {lambda5,Lambda5Input},
 {mEt2, mEt2Input},
 {Yn, LHInput[Yn]},
 {Mn, LHInput[Mn]}
};

DEFINITION[MatchingConditions]=
{{v, vSM},
 {Ye, YeSM},
 {Yd, YdSM},
 {Yu, YuSM},
 {g1, g1SM},
 {g2, g2SM},
 {g3, g3SM}};
```

```
ListDecayParticles = {Fu,Fe,Fd,Fv,VZ,VWh,hh,etR,etI,etp,Chi};
ListDecayParticles3B = {{Fu,"Fu.f90"},{Fe,"Fe.f90"},{Fd,"Fd.f90"}};
```