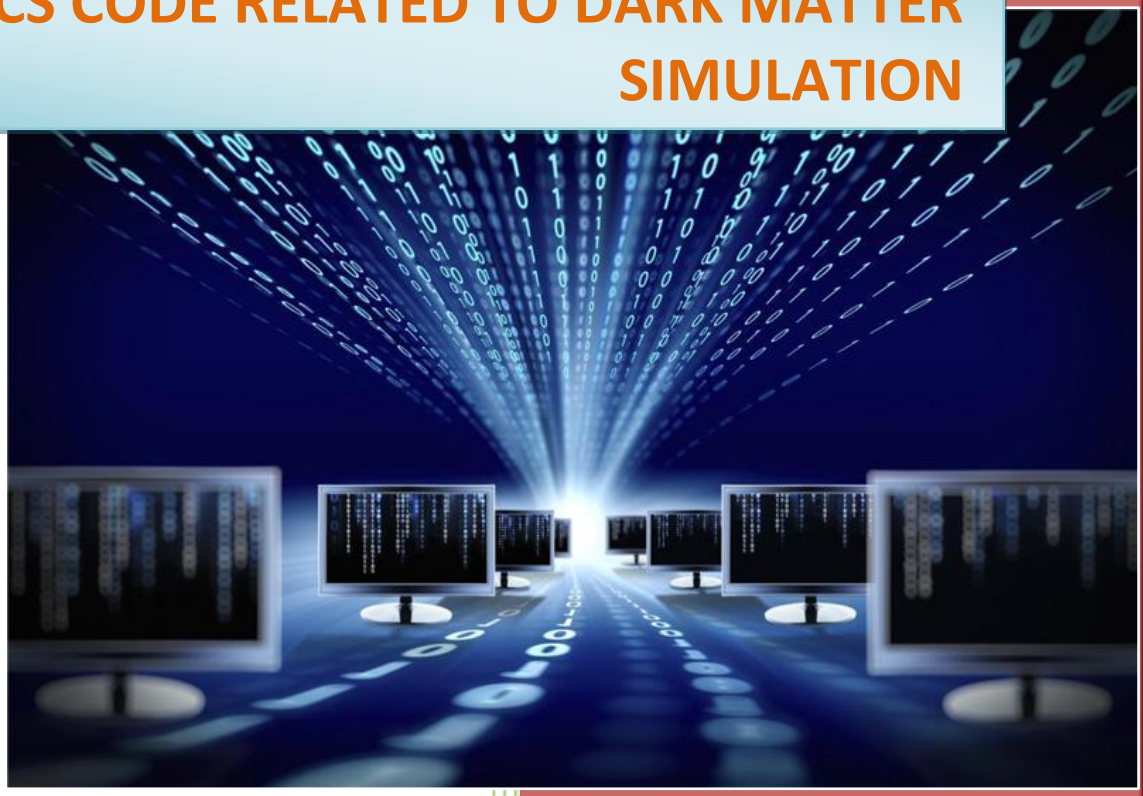
SRS

SOFTWARE FOR PARALLELIZATION OF HIGH ENERGY PHYSICS CODE RELATED TO DARK MATTER SIMULATION



Prepared for

Dr. Subhendu Rakshit

Discipline of Physics IIT Indore

Revision History

Date	Description	Author	Comments
17/1/2015	Document created	Ketan Chaware & Vishwajeet Singh Thakur	
19/1/2015	Changes in section 1.2, 3.2, 3.3	Vishwajeet Singh Thakur & Prateek Gupta	
20/1/2015	Changes in section 2.2 and addition of 3.4	Ketan Chaware & Prajwal Kinake	
21/1/2015	Addition of appendices	Prateek Gupta & Prajwal Kinake	

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Name	Signature	Title	Date

Table of Contents

1. Introduction:

Dark Matter is one of the important fields in 'High Energy Particle Physics'. Calculation of most of the properties of dark matter like Relic Density, Particle Flux, Particle Spectrum, etc. is tedious job. There are various software programs for computation of these properties. One such open source software is micrOMEGAs. This software takes huge time to calculate these properties for array of input. Our project is all about developing new alternative to this software for better performance and providing basic GUI.

1.1 Purpose:

Purpose of this SRS is introducing the original software, difficulties with it, requirements of user and our solution. Prerequisites for understanding this SRS are *very basics* of computer science and high energy particle physics.

1.2 Scope:

Software available for computation of dark matter properties mentioned above take a lot of time for computation. When a large array of input is given to them, they would take hours for computation. This really annoys the user. The problem arises because these software are running serially. Our project aims at developing new software with same functionalities and reducing computation time taken, by parallelization and optimization of code. First our team will be parallelizing the software and then we will try to optimize the code of original software, if time permits. Moreover performance of parallelized program will completely depend on specifications of users P.C.

1.3Acronyms and Abbreviations:

- SRS – Software Requirements Specification
- GUI – Graphical User Interface
- CDM – Cold Dark Matter
- IDM – Inert Doublet Model
- LHM – Little Higgs Model
- RHNM – Right-Handed Neutrino Model
- MSSM – Minimal Super-Symmetric Model
- NMSSM – Next-to Minimal Super-Symmetric Model

1.4 References:

1. micrOMEGAs website – http://lapth.cnrs.fr/micromegas/
2. Wikipedia
3. http://chpc.wustl.edu/openmp-fortran.html
4. https://computing.llnl.gov/tutorials/parallel_comp/

1.5 Overview:

Rest of the SRS contains two sections. First section is general description and enlists Product perspectives, Product functionalities, User characteristics, General constraints, Assumptions and dependencies. The second section tells about specific requirements for desired product and includes information about Interface requirements, Functional requirements and Non-functional requirements.

2. General Description:

2.1 Product Perspective:

- The product will provide simple alternative for complicated calculations in particle physics.
- Our software is supposed to be supported on LINUX.
- Developed product will be running on standalone PC and will not require network connection.

2.2 Product Functions:

Developed product will be able to compute following properties of CDM:

- Relic Density
- Particle Flux
- Particle Spectrum
- Cross Section and Decays
- Neutrino Signal from Sun and Earth

2.3 User Characteristics:

User should have basic knowledge of LINUX for using the product. Also user must have knowledge of High Energy Physics and specially Particle Physics.

2.4 General Constraints:

- Software is constrained to run only on single PC and so resources are limited.
- Also it has to be run on LINUX.
- Accuracy of output data has to be maintained at high level.

2.5 Assumptions and Dependencies:

- **User has got P.C. with multi-threading or hyper-threading.**

While parallelizing at software level, program is divided into multiple threads which run in parallel only if processor is multi-threaded. Otherwise threads run serially and parallelization is of no use.

- **User has got P.C. with at least two cores.**
  Parallelization also includes some other overheads. If processor has only single core then overall effect of parallelization may also increase time requirements. But if user has got multi-threaded processor with several cores then overall effect of parallelization becomes significant.

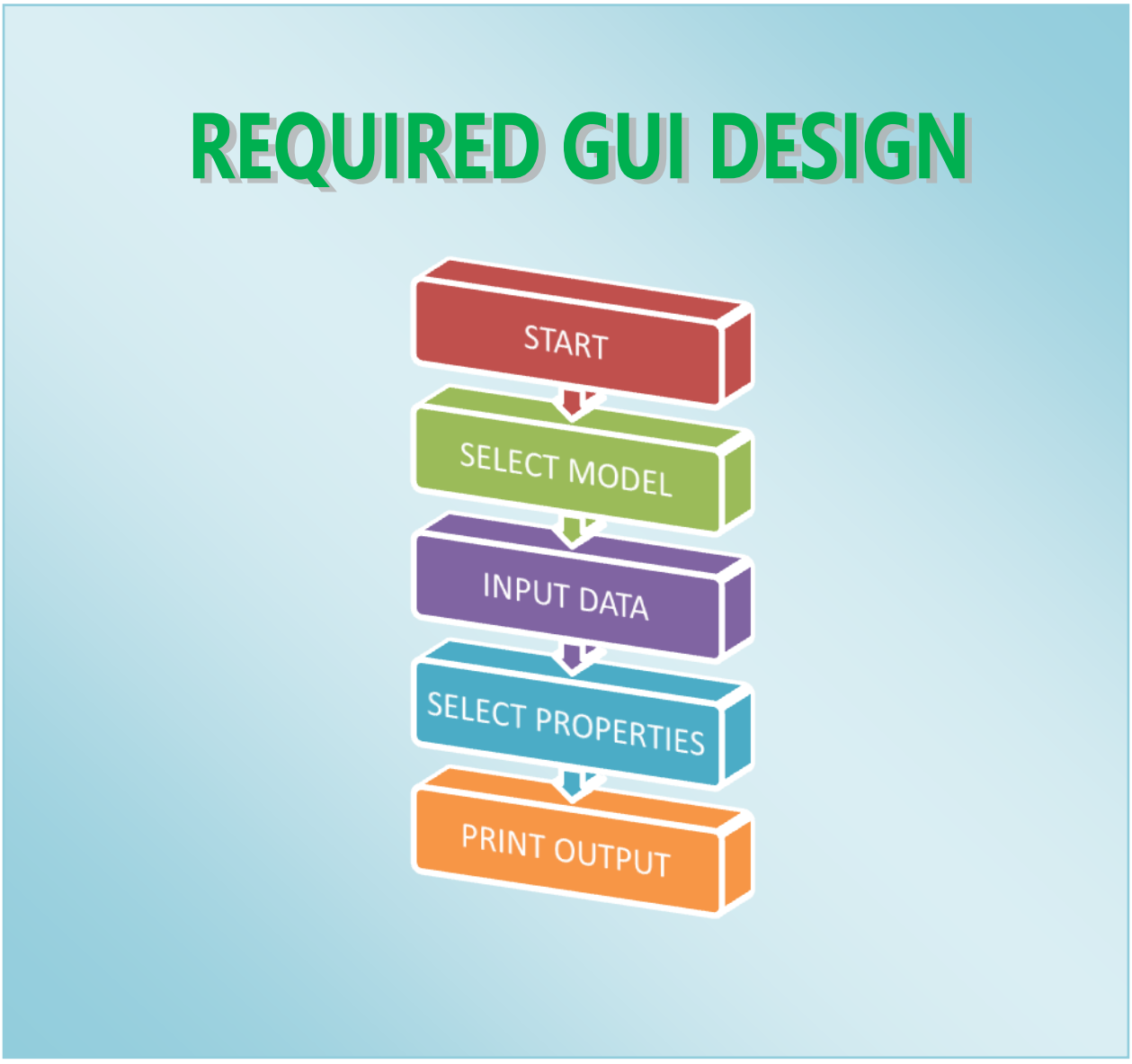
3. Specific Requirements:

3.1 External Interface Requirements:

No external hardware or software interface is required for the product.

3.2 Functional Requirements:

3.2.1 GUI Requirements:



User wants a simple GUI. After starting program it should ask for selecting particular model for computation. Following models of computation should be available:

- LHM
- IDM
- MSSM
- NMSSM
- Z4ID
- Z3M
- SM4
- RHNM
- CPVMSSM

After selection of model user should be shown default values of various parameters. User should be able to change the values of these parameters. If user wants particular parameter to vary over a range, then he should be able to set initial and final value along with step size. After this user should be asked to select properties which he wants to compute. Following properties should be available for selection:

- Relic Density
- Particle Flux
- Particle Spectrum
- Cross Section and Decays
- Neutrino Signal from Sun and Earth

Then program should process the input data and produce desired output.

3.2.2 Processing input and providing output:

For given input parameters user wants relic density, particle flux, particle spectrum, distribution of dark matter in galaxy, cross sections decays as output. All processing programs are already present in 'micrOMEGAs'. We will use these programs after parallelizing their code to improve the performance.

3.3 Non-Functional Requirements:

3.3.1 Performance:

- Time taken by software for giving output for array of inputs should be reasonable.
- Software should optimize the use of resources by running in parallel.
- Software should be user friendly in way of giving input and getting output.

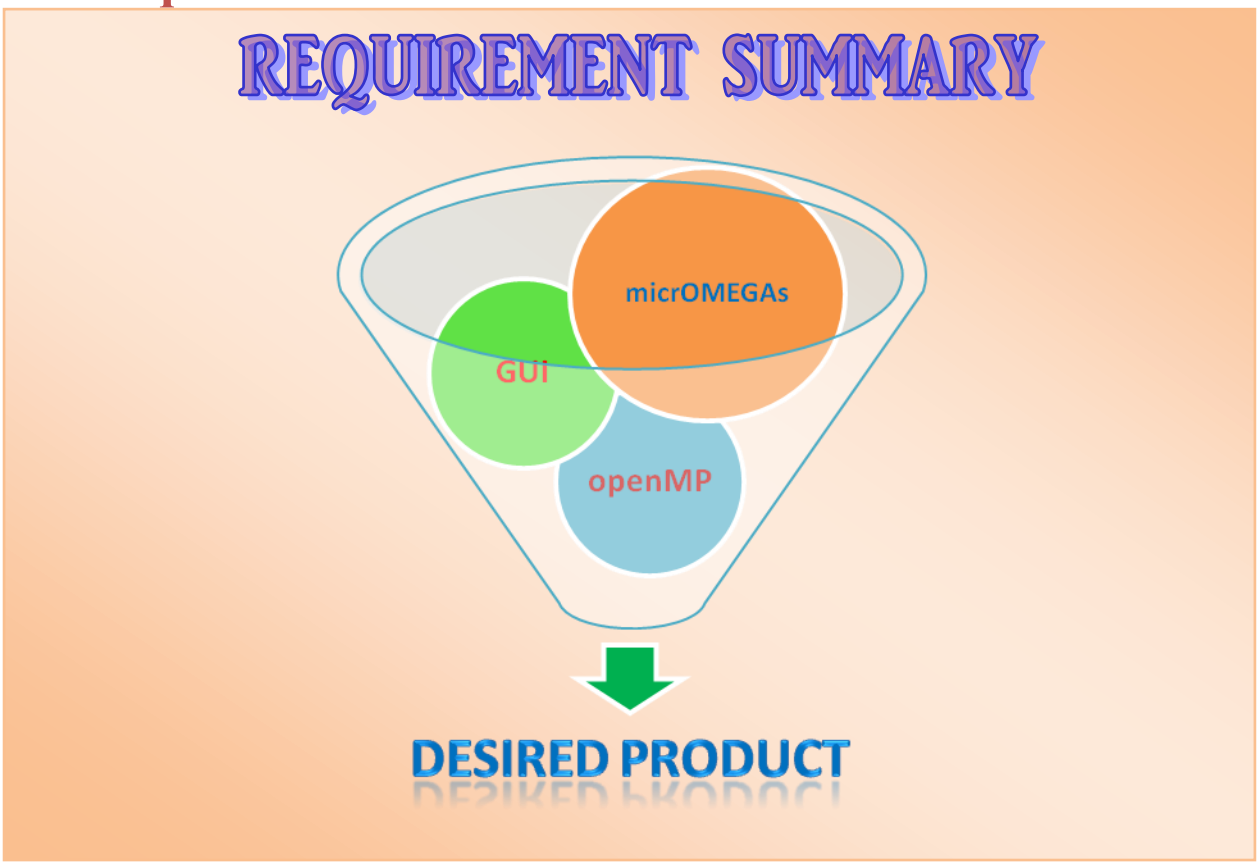
3.3.2 Usability:

GUI should be intuitive and should navigate in the most expected way.

3.3.3 Maintainability:

User wants C code of product to be well documented so that he himself will be able to change it in future to meet his changed requirements.

3.4 Requirement in nutshell:



4.APPENDICES:

A. Introduction to micrOMEGAs

1. It is a code for calculation of *dark matter* properties including the *relic density*, *direct* and *indirect rates* in a general super-symmetric model and other models of new physics.

2. It was first developed to compute relic density of a stable massive particle, now it also computes direct and indirect detection rates of dark matter.

3. The cross-sections for both spin dependent and spin independent interactions of WIMPS on protons are also computed automatically as well as the rates for WIMP scattering on nuclei in a large detector.

4. The neutrino flux and the induced muon flux from DM captured in the Sun and the Earth are also computed.

5. Annihilation cross-sections of the dark matter candidate at zero velocity, relevant for indirect detection of dark matter, are also computed automatically.

6. The decay widths of all particles in the model as well as the cross-sections for production of any pair of new particles at colliders are also computed automatically

Present version is 4.1.5

B. Introduction to openMP :

❖ *OpenMP Is:*

- An Application Program Interface (API) that may be used to explicitly direct multi-threaded, shared memory parallelism
- ***An abbreviation for***:
  - → Short version: **Open Multi-Processing**
  - → Long version: **Open** specifications for **Multi-Processing** via collaborative work between interested parties from the hardware and software industry, government and academia.
- ***Comprised of three primary API components:***
  - → Compiler Directives
  - → Runtime Library Routines
  - → Environment Variables

❖ *Goals of OpenMP:*

→ ***Standardization:***

- Provide a standard among a variety of shared memory architectures/platforms
- Jointly defined and endorsed by a group of major computer hardware and software vendors

→ ***Lean and Mean:***

- Establish a simple and limited set of directives for programming shared memory machines.
- Significant parallelism can be implemented by using just 3 or 4 directives.

- This goal is becoming less meaningful with each new release, apparently.

→ ***Ease of Use:***

- Provide capability to incrementally parallelize a serial program, unlike message-passing libraries which typically require an all or nothing approach
- Provide the capability to implement both coarse-grain and fine-grain parallelism

→ ***Portability:***

- The API is specified for C/C++ and Fortran
- Public forum for API and membership
- Most major platforms have been implemented including Unix/Linux platforms and Windows