Getting started with IMINTDYN

This document describes how to set up, compile and run IMINTDYN on a UBUNTU LINUX computer using Intels OneAPI © Compiler with HPC toolkit.

We run the program on two DELL desktop workstations

(1) DELL Precision Tower 7865 with AMD Ryzen Threadripper Pro 7865 , 24 cores
(2) DELL Precision Tower 7875 with AMD Ryzen Threadripper Pro 7875 , 24 cores

Operating System: UBUNTU Linux 24.04 ©
Compiler: Intel OneAPI © compiler with HPC toolkit

The duration of a simulation, even a complex simulation for RBS, Sputter Erostion, Coincidence ERDA, SIMS or LEIS, is typically much faster than performing the corresponding experiment.

IMINTDYN provide a huge spectrum of output data, all in form of ASCI data files. It is therefore recommended to install sufficient hard disk space of several TBytes.

The ASCII files can be easily read into Graphics and Data processing programs such as ORIGIN ©. Therefore we do not provide any specific graphics programs for plotting data from INIMTDYN.

General program structure

The imintdyn software must be copied into the home directory. For the current version a folder inimtdyn82 and several subfolders are created. The folder structure of IMINTDYN is showen in the diagram in Figure 1.Figure 1

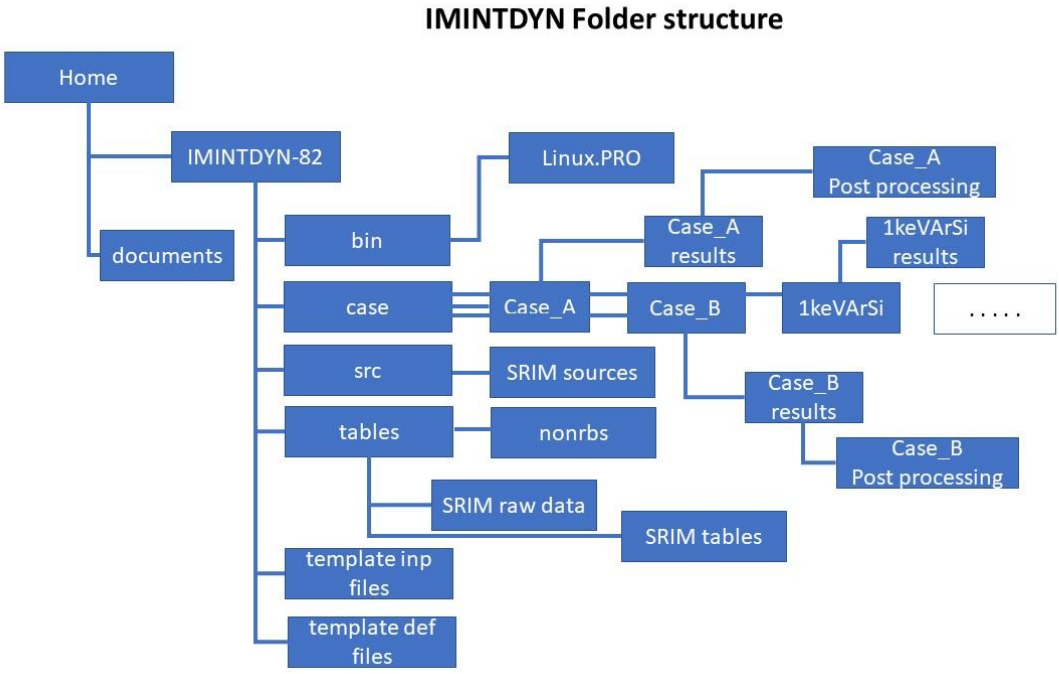

Figure 1: Folder Structure of IMINTDYN, here shown for Version 8.2

Documents: The default folder documents will contain the IMINDYN manuals, command parameter files and relevant literature and FORTRAN90 documents

Bin: The compiled program and the .obj files are available in the subfolder linux.PRO. The label PRO comes form the SDTrimSP program and describes the parallel processing version of the program.

Important: the folder linux.PRO contains a file named “hostfile” this file contains the computer names which are used to run IMINTDYN in parallel mode. Typically it contains the name of the computer used, but it can also be extend to other computers connected together as a computer cluster.

When compiling the program, the F90 source program in subfolder src are used. Compilation is done using the makefile command “sh mk”. Mk will copy the file Makefile from the src directory to linux.PRO. Details of the compilation, in particular compiler commands are specified in Makefile.

Src: This folder contains the F90 sources of IMINTDYN as well as the compilation description in Makefile.
The subfolder SRIM_sources contains a program read_srim_stopping which reads the SRIM raw stopping data and updates the stopping data files in subdirectory SRIM tables. Since the stopping data are complete this program is usually not needed.

Tables: Here are tables providing I MINTYN with element specific or compound specific data.

- Table-elements data of target and projectiles elements
- Table-compounds data for binary compound
- Table-dimer data for compounds relevant for dimer and trimer sputtering
- Table Hestopping1 stopping data for Helium projectiles part 1
- Table-Hestopping2 stopping data for Helium projectiles part 2
- Table Hstopping stopping data for Hydrogen projectiles
- Table isotopes data for isotopes of the elements
- SCOEF95A,SCOEF95B data for Ziegler Biersack stopping
- SRIM tables compressed stopping data tables used in IMINTDYN
- SRIM raw data raw stopping data tables extracted from SRIM2013
- Nonrbs non-RBS cross section data from the IBANDL data base
- New tables Excel files of the elements and compounds data

Template inp files: This folder contains a number of examples for various simulations. The script files can be used as templates to create new simulation script files

Template def files: This folder contains a number of examples for various sample and projectile definition files, such as layer profiles, projectile angular distributions, projectile energy distribution etc. . The script files can be used as templates to create new simulation script files.

Case: This folder typically contains a number of subfolders, each of them describing a particular simulation. The subfolders named case_a, case_B and 1keVArSi in Figure 1.

A cade-folder initially contains the simulation script files

- imint-caseA.inp main input script file example: imint-ArSi.inp
- caseA-modification.inp secondary input script file example: 2keV.inp
- imint.sh
- layer-definition.def and/or angle-definition.def and or energy definition.def, …

In the above example a simulation of Ar ions into a Silicon target is specified. The corresponding energy is specified in the secondary input file. Several of main and secondary input files can be specified and are eventually defined in the inim.sh script file

After a simulation is executed, subfolders with the results and the post-processing results are created.

The imint.sh script file:
A simulation is executed using the bash-command “bash imint.sh”. The imint.sh file must be edited to get the desired simulation. This includes

- defining the number of cores and threads used
- defining the primary and secondary input files
- defining the support files with extenstion .def
- defining the post-processing program to be executed after imindyn execution is finished

The imint.sh script file contains all necessary comments to edit the file
At the beginning there are two important lines in the imin.sh file:

```
TM='-off'
#TM='-testmode' # if set, then IMINTDYN is run in testmode
```

As shown, the test mode otion is switched off and the program will be executed. If the hashtag # character is removed, the the program is executed in test mode, which means that no output data are created. The testmode is very valuable, since it allows to check if the input script files are read correctly. In case of an error, an error code is indicated and the terminal output shows in which line of the ” .inp” script files a command is not correct. One should use the testmode, until an error code 0 is achieved.

During execution of a program the data are stored in a temporary directory temp-caseA in the case-folder. After execution the results are copied to a directory in subfolder caseA and the temporary directory is deleted.
In testmode, data are stored in the directory temp-testmode. In case of an error, the data there can be checked, for example to logbook file of the simulation.

Setup of the LINUX PC for Compiling IMINTDYN

Install the OneAPI Compiler including the HPC toolkit according to the Intel compiler installation guide

After installation of the OneAPI Compiler and the HPC toolkit :

1. go to the “HOME” folder. Press CTRL H to show or hide the hidden system files in this folder

2. Edit the .bashrc file by adding the following lines at the end of this file

   ```
   PATH=$PATH:opt/intel/oneapi/mpi/latest/bin
   source /opt/intel/oneapi/setvars.sh
   ```

When you then open a terminal window, the following script loads the path variables and initializes OneAPI

```
: initializing oneAPI environment ...
   bash: BASH_VERSION = 5.0.17(1)-release
   args: Using "$@" for setvars.sh arguments:
:: advisor -- latest
:: ccl -- latest
:: compiler -- latest
:: dal -- latest
:: debugger -- latest
:: dev-utilities -- latest
:: dnnl -- latest
:: dpcpp-ct -- latest
:: dpl -- latest
:: ipp -- latest
:: ippcp -- latest
:: mkl -- latest
:: mpi -- latest
:: pti -- latest
:: tbb -- latest
:: umf -- latest
:: vtune -- latest
:: oneAPI environment initialized ::
```

3. Before you can compile the IMINTDYN program, you must install the Gnu Compiler Collection GCC. This is required by the latest Intel Compiler.

You will get an error message during compilation it the package is not installed but required.

To install the GCC package one has to run the following Script once:

see the document: software installation - How to install gcc-14 on Ubuntu 22.04 and 24.04_ - Ask Ubuntu.pdf

```
sudo apt install build-essential
sudo apt install libmpfr-dev libgmp3-dev libmpc-dev -y
wget http://ftp.gnu.org/gnu/gcc/gcc-14.1.0/gcc-14.1.0.tar.gz
tar -xf gcc-14.1.0.tar.gz
cd gcc-14.1.0
./configure -v --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-
linux-gnu --prefix=/usr/local/gcc-14.1.0 --enable-checking=release --enable-
languages=c,c++ --disable-multilib --program-suffix=-14.1.0
make
sudo make install
```

note: the three lines beginning with ./configure are actuall one command line !!

The execution will take some time, so don't worry and wait until all commands have been completed.

4. Then you can open a terminal and go to the IMINTDYN folder
   cd imintdyn82/bin/linux.PRO/

5. In this folder is a file "hostfile", which contains the computers name
   Example: aghofsaess-Precision-7875-Tower
   This name can be extracted by opening the "settings" of Ubuntu Linux and open the folder "system". The compiler uses this host name to identify the computer (or computers) which is used for parallel processing. Several host names can be specified for parallel computing, but then they must be connected through the network. Details for running IMINTDYN on a computer cluster can be found in the HPC toolkit description of the OneAPI compiler.

6. **Start the compiling IMINTDYN using the script sh mk**
   You mus be in the folder: imintdyn82/bin/linux.PRO/

7. **Go to the directory "case"** and select a folder for a specific simulation. A simulation is defined by
   (1) a script file imint-name0.inp
   This main script file contains the common commands and specification of a simulation
   (2) optional script files namex.inp
   These script files contain variations of the main script file, such as different incidence angles of different ion energies, or different surface curvatures
   (3) an optional script file name-layer.def, name-angle.def, name-energy.def
   These script files describe a layered target structure, a distribution of incidence angles or a distribution of incidence energies.
   (4) a script file imint.sh or imint-name0.sh ,…

   This script file compiles all the information for a simulation and defines which post processing programs are executed. This script file uses **bash commands** !
   In the first lines of the script file you find

```
TM='-off'
#TM='-testmode' # if set, then IMINTDYN is run in testmode
```

If you eschange the “#” character so that

```
#TM='-off'
TM='-testmode' # if set, then IMINTDYN is run in testmode
```

Then the program runs in test mode. No simulation is executed but in test mode you can evaluate if all the script files were read correctly. You will get error messages in which line of the imint.inp file something is wrong specified.

A simulation is started by: bash imint.sh or bash imint-name0.sh

The advantage to use a specific name for the imint.sh file is, that the temporary data are stored in specific temporary directories. In this case an can run for example two different simulations in parallel using 16 cores each on a 32 core processor.