सत्यमेव जयते

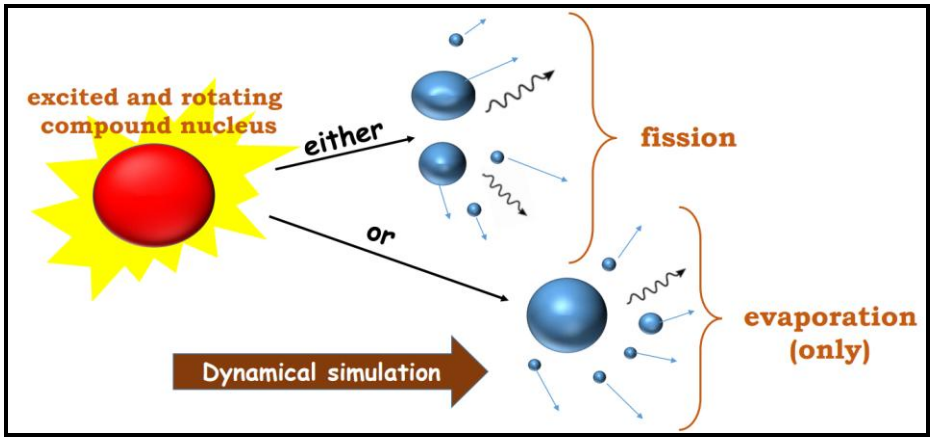
Government of India
Department of Atomic Energy

VARIABLE ENERGY CYCLOTRON CENTRE

MANUAL

Of

Nuclear Reaction Code VECLAN1D (v1.6)



Primary Developers: Santanu Pal*, Gargi Chaudhuri, and Jhilam Sadhukhan

Nuclear Dynamics and Nuclear Astrophysics Division
Theoretical Nuclear Physics Group

Variable Energy Cyclotron Centre

1/AF BIDHANNAGAR, KOLKATA – 700 064

***Superannuated**

PREFACE

Although nuclear fission has been discovered more than eighty years ago, this topic is still an open field of research mainly due to the lack of understanding of the inter-nucleonic interactions. Also, fission is an important nuclear decay mode because of multifaceted applications ranging from basic science to nuclear engineering. Particularly, fission plays a critical role in astrophysical nucleosynthesis process and in the production of superheavy elements. These subjects renewed the interests in fission research and associated nuclear reaction studies, where the entrance channel can vary from light-ion induced reactions to heavy-ion reaction.

Different dynamical effects such as nuclear dissipation, deformation-dependent shell correction, etc. are crucial while studying the heavy-ion induced reactions. The present code is intended to simulate the dynamical evolution of excited compound nuclei starting from the compact ground-state deformation till the time when either the fission-fragments or evaporation residues are formed. Three major advantages of the present code with respect to the freely available statistical model codes are: (i) it includes all the available dynamical features, (ii) different decay channels can be selected flexibly depending on the users' requirement with no prerequisite on the data file, (iii) different time distributions can be generated as the code can follow the real-time dynamics even up to a very large time reaching the upper limit for the fission decay process. Apart from these, the technical advantage is that the code can be run in parallel mode using MPI based parallelization. It substantially reduces the computational time.

The current code is implemented with the elongation degree of freedom that restrict the calculations only to the symmetric fission fragments. Therefore, the code is more suitable to apply at higher excitation energies where symmetric fission dominates. Nevertheless, for studies which are not directly affected by the fission fragment properties, the code can be used safely without any limitations on the excitation energy.

The code was developed at VECC, Kolkata, and it is improved continuously over a long period time. It helped in producing three thesis works and several highly sited journal papers. Apart from pure theoretical works, the code can serve as the workhorse for the data analysis in the domain of low-to-medium-energy nuclear reactions planned at different accelerator laboratories. The key persons responsible for the implementation of VECLAN1D v1.6 are Dr. Santanu Pal, Dr. Gargi Chaudhuri, and Dr. Jhilam Sadhukhan. Apart from them, Dr. M. T. Senthilkannan and Dr. Tathagata Banerjee have participated in various respects.

Code VECLAN1D – Version 1.6

CONTENTS

1.0 Introduction

In heavy-ion-induced reactions, an excited compound nucleus (CN) is formed when the target-projectile combination maintains a suitable dynamical condition described by their mass asymmetry, centre-of-mass energy, and angular momentum. Compound nuclear reactions are routinely studied theoretically and experimentally in different particle accelerator laboratories. The primary objective of these studies is to understand the static and dynamic properties of nuclei under moderate to extreme conditions of thermal excitation and angular momentum. Generally, different fission and evaporation residue yields and their angular distributions are measured to extract the required information.

VECLAN1D is a FORTRAN90 code that explores the evolution of excited compound nuclei by simulating their real-time dynamics as described by the one-dimensional stochastic Langevin equation. Here, nuclear elongation is used as the dynamical coordinate. VECLAN1D performs Monte-Carlo-based event-by-event simulation either sequentially or in parallel mode using the MPI architecture. It checks for the emission of light particles (n, p, α) and γ-rays at each time step of the collective evolution. The dynamics is followed up to a predefined maximum dynamical time. If an event is not completed within this time, then the faith of that event is decided by a statistical model calculation. The typical runtime of the code strongly depends on the properties of the CN (mass, shell correction, temperature, etc.), maximum dynamical time, and the number of events demanded.

VECLAN1D can simulate different probabilities such as fission and evaporation residue (ER) cross-sections and particle multiplicities, e.g. average neutron, proton, α, and γ-ray multiplicities and their energy spectra in both fission and ER events. Moreover, it can predict different time scales, evaporation yields from different deformations, partial yields from different decay channels, total kinetic energy of fission fragments, etc. However, being a one-dimensional code, VECLAN1D is restricted to symmetric fission only, and the angular distribution of observables cannot be calculated.

2.0 Decay algorithm & flow chart

The details can be found in [1]. After forming a fully equilibrated CN, the decay occurs following two distinct routes. The CN may undergo fission, where two heavy fragments are formed. During fission, the intermediate system may evaporate light

particles (n, p, α) and γ-rays until the scission configuration is reached. These evaporated particles are called pre-scission particles. After scission, the heavy fission fragments (FFs) are generally excited enough to evaporate light particles and γ-ray. These are called post-scission emissions. It is experimentally possible to distinguish between the pre- and post-scission emissions. Along the second decay route, the excitation energy of the CN is removed solely by the evaporation of light particles and γ-rays, and it stops when the excitation energy drops below the evaporation threshold. The deexcitation of the system thus ends with the formation of an ER. A schematic diagram for the decay of a CN is shown in Fig. 1. For γ-rays, the emission process continues until the Yrast-line is reached.

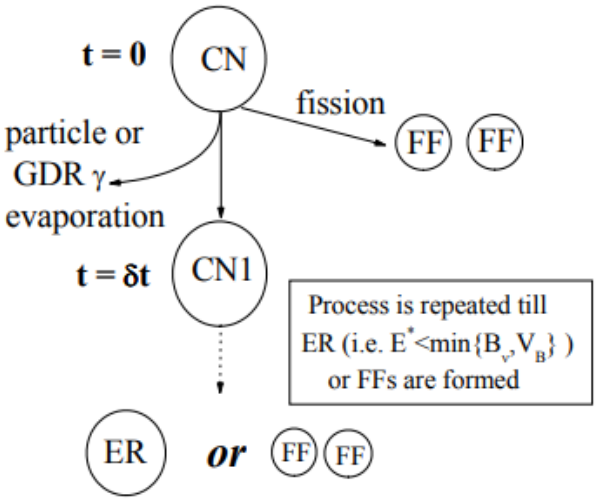


Figure 1: A schematic diagram of the decay of an excited CN

During the formation process of the compound system, some light particles can also be emitted, which may become significant with increasing bombarding energy. These particles are called pre-equilibrium particles. Since VECLAN1D starts after the formation of a CN, it does not take into account these pre-equilibrium particles. In heavy-ion induced reactions, the dinuclear system may decay without forming a CN. Such a process is categorized as quasi-fission, which is not accounted by the present code.

VECLAN1D is a combined dynamical+statistical model (CDSM) code, where the user can adjust the maximum dynamical time for an event. At the beginning of a run, one needs to calculate the following input quantities for the CN and all required daughter nuclei: (1) potential energy profile, (2) collective inertia, (3) dissipation strength (in the case of shape-dependent dissipation), (4) shell correction and level densities, and (5) evaporation widths. The technical details for running the code are

described in a subsequent section and the MANUAL. In brief, the flow chart of the dynamical part is drawn in Fig. 2. Dynamics is followed till the scission deformation is reached or the excitation energy is reduced below the particle evaporation threshold. If an event is not decided within the dynamical time t_{dyna}, then the code initiates a statistical decay algorithm as given in Fig. 3

For each event, initial angular momentum is sampled from the fusion spin distribution, which is either provided externally or calculated from systematics. Also, at each time step of the dynamics, the possibility of evaporation and, subsequently, the evaporation channel (if any) must be sampled. These random sampling are implemented as follows. Once the emission widths are known at a time t, the code calculates the ratio $x = \delta t/\tau_{tot}$, where τ_{tot} being the total decay time for neutron, proton, α, and γ decay and δt is the dynamical time step equals 0.0005 $\hbar/MeV$. Since δt is small, the probability of evaporation is given by $P(t) = 1 - \exp\left(-\frac{\delta t}{\tau_{tot}}\right) \approx x$. We then choose a random number, r_1, by sampling it from a uniformly distributed set between 0 and 1. If we find $r_1 < x$, it is interpreted as an emission process during that interval. Since the time step δt is chosen sufficiently small, the probability of decay is small, and it guarantees that in each time interval, more than one decay process is prohibited. If decay occurs within δt, then a particular decay channel is selected by another Monte-Carlo sampling. Otherwise, the above steps are repeated in the next time step δt. The excitation energy, mass, charge, and angular momentum of the CN are recalculated after each emission, and also, the inputs (1) - (5), as mentioned above, are replaced by that of the daughter nucleus. In case of fission, decay of the excited fragments is calculated statistically. Apart from realistic calculation of compound nuclear decay, VECLAN1D can be used to simulate the Langevin dynamical fission decay width as a function of time.

3.0 Advantages and salient features

Several statistical model codes are freely available to analyze fusion-fission and fusion-evaporation reactions. For example, codes named CASCADE, JOANNE,

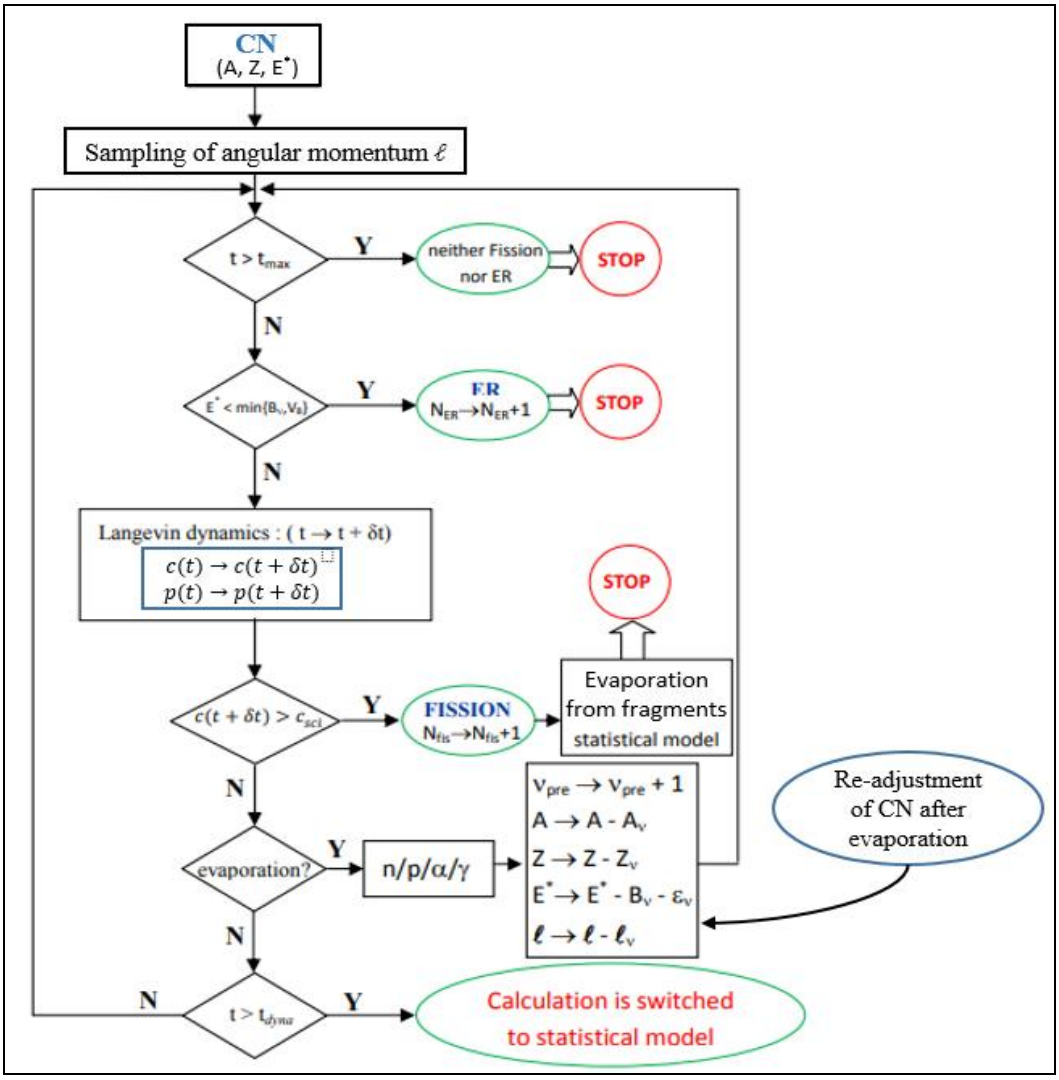


Figure 2. Langevin dynamical decay algorithm for compound nuclear decay.

TALYS, ABLA, PROFI are globally used. These codes are either purely statistical in nature, i.e. no time dependence is incorporated, or mimic dynamics through some time-dependent parametrized form of the fission width. Generally, the effect of dynamics may remain marginal at low excitation energies and, therefore, the above mentioned codes perform quite efficiently in this domain. However, with increasing excitation energy, dynamical effects become dominant and non-linear in nature such that an actual dynamical evolution is necessary. VECLAN1D can serve this purpose as it includes several unique features such as (1) shape dependent shell corrections and level densities, (2) shape-dependent dissipation strength, (3) accounting of all kinds of dynamical effects that may appear due to deformation, (4) flexibility to identify near-scission and off-scission particle emissions, (5) tunable scission condition that helps to reproduce the fragments' kinetic energy, (6) adaptivity to add or remove decay channels according to the requirements, (7) accurate prediction of

fission time and the effects due to multi-chance fission, (8) simulate different kinds of time distributions.

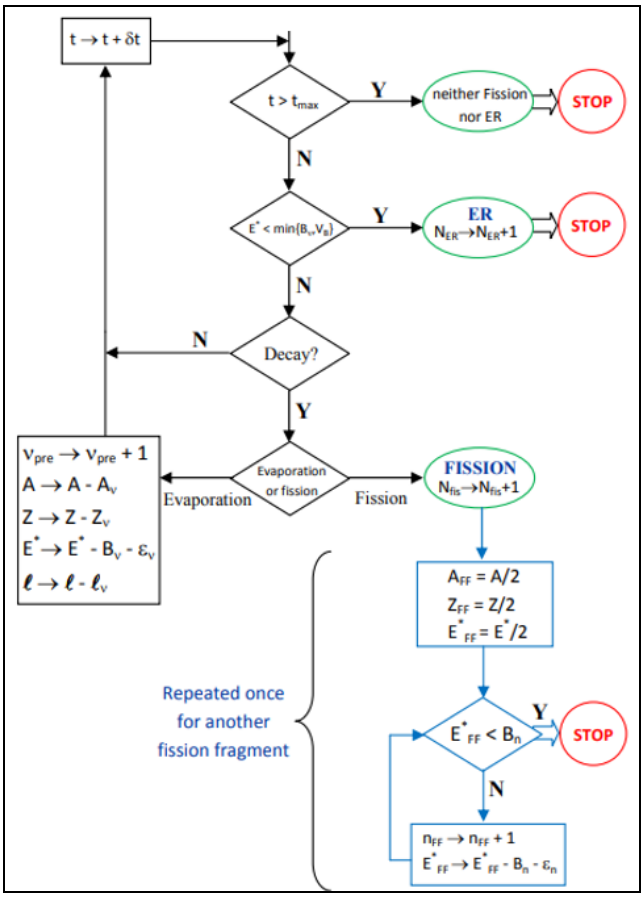


Figure 3. The Statistical decay algorithm for compound nuclear decay.

Apart from the physics advantages, VECLAN1D is written in a modular form where local changes such as modifications in the potential calculation, prescription for the particle evaporation widths, and other input quantities can be implemented easily without disturbing the main program.

4.0 Limitations

The major disadvantage of VECLAN1D is that the code considers only elongation as the dynamical coordinate. Therefore, several multidimensional effects can not be evaluated. For example, fission fragment yield distributions cannot be calculated as VECLAN1D simulates only mass symmetric fission pathways. Certain nuclei like actinides undergo asymmetric FF mass distribution at low excitation energies, and, hence, VECLAN1D should be applied carefully to avoid such situations. Also,

quantities like mass gated neutron multiplicity cannot be calculated using VECLAN1D. Another dynamical coordinate important for fission dynamics is the neck degree of freedom that may influence the FF kinetic energy. Although VECLAN1D predicts the FF kinetic energy, finer corrections due to the neck cannot be incorporated. A two-dimensional Langevin dynamical code, which will be capable of predicting the FF mass yields, is under development at VECC, Kolkata.

5.0 How to run

Following steps illustrate how to run the code VECLAN1D. The folder 'veclan1.6' contains the source code and necessary inputs.

Step 1

To compile:

(1) cd to the folder 'veclan1.6'

(2) open the 'Makefile' and prepare it as follows:-

For sequential run use:

```
#COMPILER       = IFORT
#FORTRAN_MPI    = mpiifort
COMPILER        = GFORTRAN
FORTRAN_MPI     = gfortran
USE_MPI         = 0
```

For parallel MPI run use (may be computer specific):

```
COMPILER        = IFORT
FORTRAN_MPI     = mpiifort
#COMPILER       = GFORTRAN
#FORTRAN_MPI    = gfortran
USE_MPI         = 1
```

(3) Write 'make' in the command prompt and run. This will create the executable 'veclan1d1.6'.

Note: Preprocessing attributes may require to be changed depending on the computer.

Step 2

To run the code:

(1) Create a folder (for example: 'run1') to run the code

(2) Move the executable 'veclan1d1.6' inside 'run1'

(3) Create folders 'input' and 'output' in 'run1'

(4) Copy 'langpa.in', 'mass-table.in', 'data2.in', and 'frcaos.in' in 'input'
(5) Modify 'langpa.in' as required (check MANUAL for details)
(6) Run the executable. For parallel run one example pbs script is given 'example.pbs'
(7) Output file 'lan.out' will be generated in the folder 'output'

The Makefile in the current version of the code is prepared for either the GNU compiler GFORTRAN (sequential run) or the Intel compiler IFORT (MPI run). However, the compilation with other compilers can easily be implemented.

6.0 Preparing the input file

The screenshot of a typical input file (langpa.in) is illustrated in Fig. 4. The first column contains the keywords that identifies the type of inputs. Detailed description is given below. There is no particular sequence of input data and the data can be read in any order. A complete set of inputs should be given before 'EXE_CUTE' as this key-word initiates the program and Langevin time-evolution. Then, another set of inputs (only those quantities which are required to be redefined) can be given before the next 'EXE_CUTE'. The program is executed each time it receives an 'EXE_CUTE' and terminated once it receives 'ALL_DONE'.

```
SYS_TEPR   APROJ ATART ZPROJ ZTART
            16    186     8    74
CAL_MODE   ISTAT IFODE DYNAT
             2      0   1000000.0
POT_MODE   KEYPT NFREE NSHEL BRSCL
             1     1       2    1.0
INER_CAL   KEYIN
             1
FRIC_CAL   KEYFR MODEF SKFRI AKFRI FRICT
             1     1   0.25   1.0   5.0
WDTH_CAL   KEYWD NWDTH KCOLL ECRIT DCRIT AKFCT
             1     0       1  40.0  10.0   10.0
FUSI_EEL   NSPIN NOSPN ILCVL
             1    47     40
DECAY_CN   IFREG NONUT NCHNL
             1    10     25
ENRGY_VL   IETAG ENTEM
             3   100.0
TIME_FRM   TFORM
            0.0
PRNT_OUT    IDTBN IFTIM ITCOR
             1     1      0
NO_EVENT   NEVNT
           30000
SCISSION   SCICN
            0.30
REC_READ   IRECD
             0
EXE_CUTE
ALL_DONE
```

Figure 4. The screenshot of the input file.

(1) Key word: **SYS_TEPR**

This defines the target projectile combination for the reaction. The CODE can take any combination of target and projectile. 'mass-table.in' needs to be checked for the corresponding entries.

Parameters:
NAPRO = (integer) mass of the projectile
NATAR = (integer) mass of the target
NZPRO = (integer) Z of the projectile
NZTAR = (integer) Z of the target

(2) Key word: **CAL_MODE**
This defines the mode of calculation. The CODE can be used for-
(i) Pure statistical model calculation (NO Langevin dynamics).
(ii) Pure dynamical calculation (only used for get dynamical fission width, not for realistic calculation and evaporation of particles must be swithed off).
(iii) Combined statistical+dynamical model calculation. This is for any realistic dynamical calculation.

Parameters:
ISTAT = (integer) 0 (pure statistical cal.)
1 (pur dynamical cal.)
2 (combined stat+dyna cal.)
IFODE = (integer) (effective only for ISTAT = 0 or 2)
0 (exponential form of Kramers' fission width is used in stat. part [2])
1 (integral form of Kramers' fission width is used in stat. part [2])
DYNAT = (real) maximum dynamical time for ISTAT = 1 or 2. For ISTAT = 2. Switches to stat. part after this.

(3) Key word: **POT_MODE**
Defines the driving potential and calculates if required. Either potential energy or Helmoltz free energy is used as the driving potential with/without microscopic shell correction from the phenomenological model [3] or the two-centered shell model adapted from [4].

Parameters:
KEYPT = (integer) 0 (precalculated potential from 'poten.in' is used)
1(potential is calculated for CN and all daughters using Yukawa+exponential double folding model by Sierk [1, 5]. Stored in 'poten.in'.)
NFREE = (integer) 0 (potential is used as driving potential and temperature is adjusted at each deformation.)
1 (free energy is used as driving potential and temp. is kept deformation independent with its value at the spherical configuration.)

NSHEL = (integer) 0 (shell effect is not inculded in potential as well level density parameter)
1 (phenomenological smooth shell effect is considered in both pot. And lev. density [3])
2 (shape dependent shell correction from the two center shell model adapted from [4])
BRSCL = (real) fission barrier scaling factor

(4) Key word**: INER_CAL**
Calculates Warner-Wheeler hydrodynamic inertia [6] for the dynamical part.

Parameters:
KEYIN = (integer) 0 (read the inertia from precalculated file 'inert.in')
1 (calculate the inertia and stored in 'inert.in'.)

(5) Key word: **FRIC_CAL**
Calculates and defines different friction modes. Shape-dependent friction can only be used in the dynamical part. Statistical part always uses a constant shape independent reduced friction [1].

Parameters:
KEYFR = (integer) 0 (shape dependent Wall+Window friction not calculated. read from 'frict.in' if required. Should be kept 0 if constant friction is used.)
1 (Wall+Window friction is calculated and stored in 'frict.in' [7])
MODEF = (integer) 0 (constant friction is used. KEYFR should be 0)
1 (Wall+Window friction is used)
2 (Chaos weighted wall+window friction [8] is used)
SKFRI = (real) reduction factor in wall friction in case of MODEF=1
AKFRI = (real) additional overall reduction factor in wall+window friction in case of MODEF=1
FRICT = (real) constant reduced friction in case of MODEF=0.

(6) Key word: **WDTH_CAL**
Calculates different evaporation widths (for ISTAT = 0 or 2) and fission width (for IFODE = 1).

Parameters:
KEYWD = (integer) 0 (widths will not be calculated. read from 'width.in' if required.)
1 (widths will be calculated and stored in 'width.in'.)
NWDTH = (integer) 0 (all evaporations are considered to occur at spherical shape.)
1 (shape dependent width will be used. Required large time to calculate widths. Not required for regular calculation.)

KCOLL = (integer) 0 (collective enhancement in evaporation widths will not be taken)
1 (collective enhancement in widths will be taken as given in Ref. [9])
2 (collective enhancement considered with user defined parameters E_crit, d_crit, peak value.
ECRIT = (real) value of E_crit in case of KCOLL=2
DCRIT = (real) value of d_crit in case of KCOLL=2
AKFCT = (real) peak value of collective enhancement in case of KCOLL=2.

(7) Key word**: FUSI_EEL**
Decides the fusion spin distribution from where initial spin of the CN for each event is sampled.

Parameters:
NSPIN = (integer) 0 (fixed compound nuclear spin is used for all events with value ILCVL)
1 (fusion spin distribution is decided from systematics Ref. [10])
2 (user given fusion spin distribution is used from 'spind.in'.
NOTE: this file must be arranged in the following order
1) first column - (real) center-of-mass energy for fusion
2) second column - (integer) value of spin
3) third column - (real) probability of that spin)
NOSPN = (integer) (Number of rows in file 'spind.in' in case of NSPIN = 2, otherwise ignored, cannot be blank)
ILCVL = (integer) (value of constant spin for NSPIN = 0, otherwise ignored, cannot be left blank)

(8) Key word: **DECAY_CN**
Decides the decay channels in case of ISTAT = 0 or 2. Different options are (i) standard: any number of neutron decay with a maximum of either 1 proton or 1 alpha decay per event. (ii) user defined: required decay channels can be written in the file 'decay.in'.

Parameters:
IFREG = (integer) 0 (all the decay channels are read from 'decay.in' where 1st column is daughter's mass and 2nd column is daughter's charge, both should be integers)
1 (regular decay algorithm with maximum of either 1 proton or 1 alpha, and NONUT number of neutrons)
NONUT = (integer) (number of neutron decays in case of IFREG = 1, NONUT = 0 is equivalent to ISTAT = 1, i.e., only fission width will be calculated without any evaporation)
NCHNL = (integer) (number of daughter nuclei in 'decay.in' in case of IFREG = 0, otherwise ignored)

(9) Key word: **ENERGY_VL**
Reads the energy for the reaction.

Parameters:
IETAG = (integer) 1 (energy in ENTEM will be considered as the energy in lab frame)
2 (energy in ENTEM will be considered as the energy in c.m. frame)
3 (energy in ENTEM will be considered as the excitation energy of compound nucleus)
ENTEM = (real) (value of energy for the current run)

(10) Key word: **TIME_FRM**
Reads the formation time during which the fission probability is kept zero.

Parameters:
TFORM = (real) (formation time in units of hbar/MeV)

(11) Key word: **PRNT_OUT**
Decides printout of different output quantities in the output file 'lan.out'. Without any switch, the CODE writes number of fission and ER events, corresponding probabilities, and multiplicities. Apart from these, other outputs are written depending on the following switches.

Parameters:
IDTBN = (integer) 0 (different distributions are not written)
1 (writes distribution of multichance events, distribution of average deformation, energy spectra of prescission particles, distribution of deformation at the time of evaporation, partial spin distribution, distribution of TKE and associate neutron multiplicity)

IFTIM = (integer) 0 (different time distributions are not written)
1 (writes fission time distributions and fission time distribution for different exit channels)
ITCOR = (integer) 0 (correlation between fission time and angular momentum is not written)
1 (event distribution for the correlation between fission time and angular momentum will be written)

(12) Key word: **NO_EVENT**
Defines the ensemble: number of events.

Parameters:
NEVNT = (integer) (number of events for the current run)

(13) Key word: **SCISSION**
Defines the scission criterion. The neck radius R_N at scission is a fraction of the spherical radius R_0, i.e., R_N=xR_0. It is observed that x=0.3 reproduces TKE for actinides. One can tune x that strongly affects TKE. Use x=0.3 if TKE info not available.

Parameters:
SCICN = (real) (value of x, where R_N=xR_0)

(14) Key word: **REC_READ**
For faster running of the CODE, several inputs are calculated only for the first run and these are then stored in the record file 'recrd.in'. These inputs can be read from this file if no major change is demanded in subsequent runs.

Parameters:
IRECD = (integer) 0 (all the inputs are calculated afreshed and stored in 'recrd.in')
1 (several inputs are read from 'recrd.in')

(15) Key word: **EXE_CUTE**
Execute the CODE based on the inputs received

(16) Key word: **ALL_DONE**
Terminates the CODE

7.0 Typical output

A typical output file is available in the ‘output’ folder.

8.0 Publications using the present code or its previous versions

1. Gargi Chaudhuri and Santanu Pal, Phys. Rev. C 63, 064603 (2001)
2. Gargi Chaudhuri and Santanu Pal, Phys. Rev. C 65, 054612 (2002)
3. Gargi Chaudhuri and Santanu Pal, Eur. Phys. J. A 18, 9 (2003)
4. Santanu Pal, Gargi Chaudhuri, Jhilam Sadhukhan, Nucl. Phys. A 808, 1 (2008)
5. Jhilam Sadhukhan, Santanu Pal, Phys. Rev. C 79, 064606 (2009)
6. Jhilam Sadhukhan and Santanu Pal, Phys. Rev. C 81, 031602(R) (2010)
7. Jhilam Sadhukhan, Santanu Pal, Phys. Rev. C 82, 021601 (2010)
8. Jhilam Sadhukhan and Santanu Pal, Phys. Rev. C 84, 044610 (2011)
9. Neeraj Kumar, Shabnam Mohsina, Jhilam Sadhukhan, Shashi Verma, Phys. Rev. C 96, 034614 (2017)

10. M T Senthil Kannan, J Sadhukhan, B K Agrawal, M Balasubramaniam, Phys. Rev. C 98, 021601(R) (2018)
11. Nishant Kumar, D C Biswas, T K Ghosh, J Sadhukhan, et. al., Phys. Rev. C 99, 041602(R) (2019)
12. Neeraj Kumar, Shashi Verma, Shabnam Mohsina, J Sadhukhan, et. al., Phys. Lett. B 814, 136062 (2021)
13. N K Rai, A Gandhi, M T Senthil Kannan, et. al., J. Phys. G 49, 035103 (2022)
14. G Mohanto, M T S Kannan, J Sadhukhan, B Srinivasan, Phys. Rev. C 110, 034604 (2024)
15. P Dubey, M Upadhyay, et. al., Phys. Rev. C 112, L011602 (2025)
16. P Dubey, M Upadhyay, et. al., Phys. Rev. C 113, 044602 (2026)
17. A Agrawal, M Thakur, et. al., Phys. Rev. C 113, 044618 (2026)

9.0 Acknowledgement

The shape-dependent shell correction is implemented by adopting the open-source BARRIER code of Ref. [4].

References:

[1] Jhilam Sadhukhan, Ph. D. Thesis (Homi Bhabha National Institute, 2012)(unpublished).
[2] Jhilam Sadhukhan and Santanu Pal Phys. Rev. C 82, 021601(R) (2010).
[3] W. D. Myers and W. J. Swiatecki, Nucl. Phys. 81, 1 (1966).
[4] F. Garcia, O. Rodriguez, J. Mesa, J. D. T. Arruda-Neto, V. P. Likhachev, E. Garrote, R. Capote, and F. Guzmán, Comput. Phys. Commun. 120, 57 (1999).
[5] A. J. Sierk, Phys. Rev. C 33, 2039 (1986).
[6] K. T. R. Davies, R. A. Managan, J. R. Nix, and A. J. Sierk, Phys. Rev. C 16, 1890 (1977).
[7] A. V. Karpov, P. N. Nadtochy, D. V. Vanin, and G. D. Adeev, Phys. Rev. C 63, 054610 (2001).
[8] Gargi Chaudhuri and Santanu Pal, Phys. Rev. C 65, 054612 (2002).
[9] Tathagata Banerjee, S. Nath, A. Jhingan, et. al., Phys. Rev. C 96, 014618 (2017).
[10] P. Fröbrich, I.I. Gontchar, Phys. Rep. 292, 131 (1998).