

«TalysLib v.0.1»

Содержание

1	Введение	1
2	Структура библиотеки	1
2.1	Класс Nucleus	1
2.1.1	Name, Reaction, Projectile	1
2.1.2	vector<Level> Levels	2
2.1.3	OpticalModelParameters OMPN, OMPP	2
2.1.4	Deformation	2
2.1.5	TGraph ElasticTotTalys, ElasticDirectTalys, ElasticCompoundTalys	2
2.1.6	InelasticTotTalysV, InelasticDirectTalysV, InelasticCompoundTalysV, ElasticTotTalysV, ElasticDirectTalysV, ElasticCompoundTalysV, TotTalysV	2
2.1.7	BNECS_gamma, BNECS_neutron, BNECS_proton, BNECS_deuteron, BNECS_triton, BNECS_3He, BNECS_alpha	3
2.1.8	TEISGraphTot, TEISGraphCont, TEISGraphDiscr	3
2.1.9	void ReadElastic()	3
2.1.10	TGraph* GetElasticAngularDistribution(string type="Total string option=)	4
2.1.11	int WriteOMPOrUseKoningP, WriteOMPOrUseKoningN	4
2.1.12	void MergeLevels(float tolerancy)	4
2.1.13	SortingLevels()	4
2.1.14	FindProductsByReaction(string reaction)	5
2.1.15	FindProductByName(string _Name)	5
2.1.16	Nucleus* fMotherNucleus	5
2.1.17	Nucleus()	5
2.1.18	Nucleus(string Name, string Reaction=)	5
2.1.19	Nucleus(NucleusData ND)	6

2.1.20	ReadENSDFFile(string filename=,string Nuclide=)	6
2.1.21	ReadLevelsFromTalysDatabase(string type="final")	6
2.1.22	SetProjectileEnergy(double E)	6
2.1.23	vector<Level*> GetLevelsWithCorrespondingTransitions(float Energy, float tolerancy=1,float intensity=0)	6
2.1.24	Level* FindLevelFromTalys(float Energy,SpinParity JP)	7
2.1.25	Level* FindBestLevelByEJP(float Energy,SpinParity JP,float tolerancy=1)	7
2.1.26	FindLevelByEnergy(float Energy,float tolerancy=1)	7
2.1.27	Level* FindLevelByNumber(int number)	7
2.1.28	GammaTransition* GetBestTransition(float Energy,float tolerancy=1)	7
2.1.29	vector<GammaTransition*> GetGammaTransition(float Energy, float tolerancy=1,float intensity=0)	8
2.1.30	vector<GammaTransition*> GetGammaTransitions(string option="Talys int BetterThan=100,float tolerancy=1,float intensity=0)	8
2.1.31	vector<Nucleus> Products	8
2.1.32	void GenerateProducts(string Projectile="n")	8
2.1.33	void ExecuteCalculationInTalys(string _Projectile="n")	8
2.1.34	void ReadTalysCalculationResult()	9
2.1.35	void AssignSimilarLevels(float Tolerancy=1.5)	9
2.1.36	void DrawLevelScheme(double MinTalysCS=0)	9
2.1.37	void AssignPointers()	9
2.1.38	void ErasePointers()	9
2.1.39	void AssignDeformationsToLevels()	10
2.1.40	void SetLevelDeformation(int LevelNumber,char LevT, int BandN=- 1, int BandL=-1, int NPhon=-1, int MagN=-1, vector<float> *Def=0)	10
2.1.41	void SetLevelDeformation(double LevelEnergy,char LevT, int BandN=- 1, int BandL=-1, int NPhon=-1, int MagN=-1, vector<float> *Def=0)	10

1 Введение

Библиотека **TalysLib** представляет собой основанный на ROOT набор классов, облегчающий взаимодействие программ, создаваемых пользователем, с базами данных и результатами расчетов TALYS. Библиотека не требует вмешательства в исходный код TALYS, и предполагается, что это позволит обеспечить хорошую совместимость с разными версиями TALYS. Для получения параметров ядра (деформаций, оптических потенциалов, масс, наборов уровней с соответствующими квантовыми числами) производится чтение базы данных TALYS (расположена в директории structure), результаты расчета извлекаются из выдачи TALYS (обычно, перенаправляемой в файл output) путем поиска ключевых слов и считывания соответствующих значений.

2 Структура библиотеки

Библиотека состоит из набора классов, описывающих различные физические свойства ядер.

2.1 Класс Nucleus

Класс Nucleus включает в себя набор объектов и методов для работы со свойствами ядра. Описание методов и членов этого класса приведено ниже

2.1.1 Name, Reaction, Projectile

string Name-имя ядра вида `"56Fe"`. string Reaction-реакция, приводящая к образованию данного ядра, строка вида `"(n,p)"`. string Projectile-налетающая частица, строка вида `"n"`. Возможные значения данного параметра-n, p, d, t, h, a, g.

2.1.2 vector<Level> Levels

Вектор, содержащий объекты типа Level-информацию о свойствах ядерных уровней.

2.1.3 OpticalModelParameters OMPN, OMPP

Объекты, содержащие в себе информацию об оптическом потенциале для нейтронов (OMPn) и протонов (OMPP)

2.1.4 Deformation

Объект, содержащий в себе сведения о деформации ядра в основном и возбужденных состояниях, а так же о природе этих состояний.

2.1.5 TGraph ElasticTotTalys, ElasticDirectTalys,ElasticCompoundTalys

Графики углового распределения упруго рассеянных частиц: ElasticTotTalys-полное сечение, ElasticDirectTalys-прямая компонента упругого рассеяния, ElasticCompoundTalys-компаунд-компонента. Данные графики заполняются функцией Nucleus::GetElasticAngularDistribution(stype,string option) из векторов Angle, ElTot; Angle, ElCompound; Angle, ElDirect соответственно.

2.1.6 InelasticTotTalysV, InelasticDirectTalysV,InelasticCompoundTalysV, ElasticTotTalysV, ElasticDirectTalysV,ElasticCompoundTalysV, TotTalysV

Графики сечений в зависимости от некоторого, определяемого пользователем, параметра V. Заполняются функцией Nucleus::AddPoint(double x_value, Nucleus* Nucl) из значений TotInelastic, CompoundInelastic, DirectInelastic, TotElastic, CompoundElastic, DirectElastic, TotTalys соответственно. Величины TotInelastic, CompoundInelastic, DirectInelastic,

TotElastic, CompoundElastic, DirectElastic, TotTalys, в свою очередь, считываются непосредственно из output файла функцией Nucleus::ReadElastic().

2.1.7 BNECS_gamma, BNECS_neutron, BNECS_proton, BNECS_deuteron, BNECS_triton, BNECS_3He, BNECS_alpha

Данные графики представляют собой полные сечения рождения частиц γ , n, p, d, ^3H , ^3He , α соответственно, в выходном файле данные величины названы «Binary non-elastic cross sections (non-exclusive)». Заполнение этих графиков производится функцией Nucleus::AddPoint(double x_value, Nucleus* Nucl) из величин BNECS_g, BNECS_n, BNECS_p, BNECS_d, BNECS_t, BNECS_tau, BNECS_a соответственно, а они, в свою очередь, считываются из output файла функцией Nucleus::ReadElastic().

2.1.8 TEISGraphTot, TEISGraphCont, TEISGraphDiscr

Данные графики представляют собой сечения неупругого рассеяния, т.е., реакций вида $(n, 1nx)$: полное, в континууме и на дискретных уровнях соответственно, в выходном файле данные величины названы «Total exclusive Inelastic scattering». Заполнение этих графиков производится функцией Nucleus::AddPoint(double x_value, Nucleus* Nucl) из величин TEISTot, TEISCont, TEISDiscr соответственно, а они, в свою очередь, считываются из output файла функцией Nucleus::ReadElastic().

2.1.9 void ReadElastic()

Функция, считывающая output файл и записывающая в переменные TOTGamProd, TOTNProd, TOTPPProd, TOTDProd, TOTAProd полные сечения образования γ , n, p, d, α соответственно; в TotElastic, CompoundElastic, DirectElastic записываются сечения упругого рассеяния и вклады отдельных компонент, а в TotInelastic, CompoundInelastic, DirectInelastic - сечения неупругого рассеяния. В переменную TotTalys записываются значения полного сечения.

2.1.10 TGraph* GetElasticAngularDistribution(string type="Total string option=)

Данная функция строит угловое распределение рассеянных нейтронов и возвращает указатель на соответствующий график. Переменная type задает возможные варианты графика. Возможные значения: "Total"полное сечение, "Compound"компаунд-компонента, "Direct"прямая компонента. Аргумент option принимает 2 значения: (по умолчанию) и "new". В первом случае функция вернет уже построенный график, во втором-перестроит его.

2.1.11 int WriteOMPOrUseKoningP, WriteOMPOrUseKoningN

Данные переменные задают алгоритм работы с оптической моделью для протонов и нейтронов соответственно. Если значение переменной равно -1, то в расчете будет использован потенциал "по-умолчанию либо заданный в input файле, если WriteOMPOrUseKoning=0, то будет использован потенциал, считанный из базы TALYS. Этот потенциал может быть изменен пользователем. Если в базе потенциала для данного ядра нет, будет использован потенциал "по-умолчанию". В случае, если WriteOMPOrUseKoning=1, то будет использован потенциал из базы, а если его не окажется-то потенциал Кёнинга. Если WriteOMPOrUseKoning=2, то будет использован потенциал Кёнинга вне зависимости от наличия в базе потенциала для данного ядра.

2.1.12 void MergeLevels(float tolerancy)

Функция, используемая для слияния уровней, считанных из TALYS и NDS. Использование в дальнейшем не предполагается.

2.1.13 SortingLevels()

Функция, выполняющая сортировку вектора Levels по энергии уровня.

2.1.14 FindProductsByReaction(string reaction)

Функция, выполняющая поиск продукта по соответствующей реакции и возвращающая указатель на найденный продукт. В случае, если продукт не найден, возвращает NULL.

2.1.15 FindProductByName(string _Name)

Функция, выполняющая поиск продукта по его имени и возвращающая указатель на найденный продукт. В случае, если продукт не найден, возвращает NULL.

2.1.16 Nucleus* fMotherNucleus

Указатель на начальное ядро, если рассматриваемое ядро является продуктом реакции и содержится в векторе Products. В случае, если рассматриваемое ядро является начальным ядром, он равен NULL. Устанавливается функцией AssignPointers().

2.1.17 Nucleus()

Конструктор по-умолчанию. Необходим для корректной работы динамических контейнеров.

2.1.18 Nucleus(string Name,string Reaction=)

Конструктор. Принимает как аргумент Name-название ядра, например, ^{56}Fe и Reaction-реакцию, в которой данное ядро образовалось. В случае начального ядра этот аргумент остается, по умолчанию, пустым. В процессе создания ядра данным конструктором выполняются считывание из базы TALYS величин Z, A, распространенности изотопа, схемы уровней, параметров деформации и оптических потенциалов, выполняется присваивание указателей на данное ядро для уровней, деформаций и оптических потенциалов.

2.1.19 Nucleus(NucleusData ND)

Конструктор, позволяющий создать объект Nucleus из более компактного объекта NucleusData. В силу технических ограничений, не удалось реализовать запись объекта Nucleus в .root файл, возможно, эта проблема связана с указателями. Объект NucleusData записывается в .root файл и считывается из него без каких-либо проблем.

2.1.20 ReadENSDFFile(string filename=,string Nuclide=)

Функция, считывающая ENSDF файлы и заполняющая уровни (Levels). Использование в дальнейшем не предполагается.

2.1.21 ReadLevelsFromTalysDatabase(string type="final")

Функция, считывающая параметры уровней из базы TALYS. В качестве аргумента принимает тип информации об уровнях. Возможные значения аргумента: "final" "exp" "hfb" описание приведено в TALYS manual (v 1.9), p. 137.

2.1.22 SetProjectileEnergy(double E)

Функция, устанавливающая энергию начальной частицы в МэВ.

2.1.23 vector<Level*> GetLevelsWithCorrespondingTransitions(float Energy, float tolerancy=1,float intensity=0)

Функция, выполняющая поиск уровней с гамма-переходами энергии Energy, энергия гамма-перехода может отличаться от заданной на tolerancy, интенсивность должна быть не ниже intensity. Использовалась для поиска подходящих гамма-переходов в данных, считанных из ENSDF (ну, или, NDS). Использование в дальнейшем не предполагается.

2.1.24 Level* FindLevelFromTalys(float Energy,SpinParity JP)

Функция, выполняющая поиск уровня, полученного из базы TALYS с данной энергией и спин-четностью JP. В случае успеха возвращает указатель на найденный уровень, в противном случае-NULL.

2.1.25 Level* FindBestLevelByEJP(float Energy,SpinParity JP,float tolerancy=1)

Функция, выполняющая поиск уровня, лучше всего подходящего под заданные энергии. В случае, когда есть несколько уровней в пределах tolerancy от Energy, лучшим будет признан тот, который имеет спин-четности, совпадающие с JP. В случае успеха возвращает указатель на найденный уровень, в противном случае-NULL.

2.1.26 FindLevelByEnergy(float Energy,float tolerancy=1)

Функция, выполняющая поиск уровня с энергией Energy, энергия уровня может отличаться на tolerancy от величины Energy. В случае успеха возвращает указатель на найденный уровень, в противном случае-NULL.

2.1.27 Level* FindLevelByNumber(int number)

Функция, выполняющая поиск уровня с данным порядковым номером (соответствует нумерации TALYS). Нумерация начинается с основного состояния, ему присвоен номер 0. В случае успеха возвращает указатель на найденный уровень, в противном случае-NULL.

2.1.28 GammaTransition* GetBestTransition(float Energy,float tolerancy=1)

Функция, выполняющая поиск гамма-перехода с энергией, наиболее близкой к Energy в пределах tolerancy. В случае успеха возвращает указатель на найденный уровень, в противном случае-NULL.

2.1.29 `vector<GammaTransition*> GetGammaTransition(float Energy, float tolerancy=1, float intensity=0)`

Функция, выполняющая поиск гамма-переходов с энергией, наиболее близкой к Energy в пределах tolerancy. Возвращает вектор указателей на найденные переходы.

2.1.30 `vector<GammaTransition*> GetGammaTransitions(string option="Talys int BetterThan=100, float tolerancy=1, float intensity=0)`

Функция, выполняющая поиск гамма-переходов с происхождением и надежностью, задаваемыми аргументами функции. Использование в дальнейшем не предполагается.

2.1.31 `vector<Nucleus> Products`

Вектор, содержащий объекты типа Nucleus. В случае, когда рассматриваемое ядро является начальным, содержит ядра-продукты, в случае же, когда рассматриваемое ядро само является продуктом, пуст. Заполняется функцией Nucleus::GenerateProducts().

2.1.32 `void GenerateProducts(string Projectile="n")`

Функция, выполняющая создание продуктов. Заполняет вектор Products, вызывает функции, выполняющие создание input файла для TALYS, его запуск и считывание результатов вычислений. В качестве аргумента принимает название начальной частицы.

2.1.33 `void ExecuteCalculationInTalys(string _Projectile="n")`

Функция, выполняющая создание входного файла для TALYS и запуск вычислений. В качестве аргумента принимает название начальной частицы. Во входной файл могут быть добавлены дополнительные команды, находящиеся в Nucleus::TalysOptions.

2.1.34 void ReadTalysCalculationResult()

Функция, выполняющая считывание результатов расчета TALYS. Считываются:

1. ADTot, ADDirect, ADCompound-угловые распределения неупруго рассеянных частиц и продуктов реакции и вклады прямых и компаунд-процессов
2. сечения излучения гамма-квантов

В теле данной функции выполняется приписывание гамма-переходов соответствующим уровням.

2.1.35 void AssignSimilarLevels(float Tolerancy=1.5)

Функция, выполняющая приписывание уровням близких по энергии. Была введена, чтобы перерабатывать огромное множество одинаковых уровней из ENSDF. Использование в дальнейшем не предполагается.

2.1.36 void DrawLevelScheme(double MinTalysCS=0)

Функция, отрисовывающая схему уровней на текущий Canvas. В качестве аргумента принимает сечение излучения гамма-кванта, переходы с сечением менее MinTalysCS отрисованы не будут.

2.1.37 void AssignPointers()

Функция, приписывающая указателям правильные значения. Присваивает fMotherNucleus указатель на начальное ядро, Level->fNucleus-указатель на ядро, к которому относится данный уровень. Необходимо вызывать при копировании и после окончания записи в контейнер.

2.1.38 void ErasePointers()

Функция, вызывающая стирание указателей.

2.1.39 void AssignDeformationsToLevels()

Функция, приписывающая уровням указатель на соответствующий объект LevelDeformation, содержащий информацию о деформации уровня и его происхождении. Приписывание деформаций осуществляется по номеру уровня.

2.1.40 void SetLevelDeformation(int LevelNumber,char LevT,int BandN=-1,int BandL=-1,int NPhon=-1,int MagN=-1,vector<float> *Def=0)

Функция, позволяющая установить деформацию для уровня с номером LevelNumber. В случае, если аргумент = -1, он будет пропущен при записи деформации в файл. Параметры деформации описаны в TALYS manual (v 1.9), p. 138, LevT соответствует типу уровня (type of collectivity), BandN-номер полосы (the number of band), BandL-момент полосы или мультипольность(?) (multipolarity), NPhon-число фононов (phonon number of the level), MagN-магнитное квантовое число (magnetic quantum number). Значения деформации ($\beta_2 \dots \beta_8$) передаются с помощью указателя на вектор *Def.

2.1.41 void SetLevelDeformation(double LevelEnergy,char LevT,int BandN=-1,int BandL=-1,int NPhon=-1,int MagN=-1,vector<float> *Def=0)

Функция, позволяющая установить деформацию для уровня с энергией LevelEnergy (точность задания энергии должна быть лучше 1 кэВ). В случае, если аргумент = -1, он будет пропущен при записи деформации в файл. Параметры деформации описаны в TALYS manual (v 1.9), p. 138, LevT соответствует типу уровня (type of collectivity), BandN-номер полосы (the number of band), BandL-момент полосы или мультипольность(?) (multipolarity), NPhon-число фононов (phonon number of the level), MagN-магнитное квантовое число (magnetic quantum number). Значения деформации ($\beta_2 \dots \beta_8$) передаются с помощью указателя на вектор *Def.

2.1.42 string PrintLevels()

Функция, возвращающая список уровней данного ядра в виде строки.

2.1.43 string PrintReactions()

Функция, возвращающая список продуктов и реакций, если данное ядро является начальным.

2.1.44 string ReactionToTalysNotation(char DataSelection=kExcitationCS)

Функция, генерирующая ключевое слово, по которому сведения о реакции, которая привела к рождению данного ядра, можно найти в файле. В качестве аргумента принимает тип информации, для которого должно быть сгенерировано ключевое слово. Возможные значения аргумента: kExcitationCS-сечение образования конкретного состояния, kAngularDistribution-угловое распределение частицы-продукта, kTotalInelasticCS-полное сечение образования данного продукта.

2.1.45 float GetMass()

Функция, возвращающая массу ядра в МэВ.

2.1.46 NucleusData ToNucleusData()

Функция, конвертирующая объект Nucleus в NucleusData, который может быть сохранен в .root файл.

2.1.47 void GenerateGammaSpectrum(TH1F *Spectrum, TF1* ResolutionFunction, int NEntries=100000)

Функция, позволяющая сгенерировать ожидаемый гамма-спектр на основе результатов вычислений в TALYS. Первый аргумент-указатель на гистограмму, куда

будет записан спектр, второй-указатель на функцию (TF1), описывающую разрешение детектора, третий-число событий в спектре.

2.1.48 void SetTGraphNameAndTitle(string ValName)

Функция, задающая имена и подписи осей для графиков. В качестве аргумента принимает имя переменной, которая соответствует оси x .

2.1.49 void AddPoint(double x_value, Nucleus* Nucl)

Функция, выполняющая заполнение графиков InelasticTotTalysV, InelasticDirectTalysV, InelasticCompoundTalysV, ElasticTotTalysV, ElasticDirectTalysV, ElasticCompoundTalysV, TotTalysV, BNECS_gamma, BNECS_neutron, BNECS_proton, BNECS_deuteron, BNECS_triton, BNECS_3He, BNECS_alpha, TEISGraphTot, TEISGraphCont, TEISGraphDiscr значениями, находящимися в объекте, на который указывает указатель Nucleus* Nucl, в качестве переменной x передается x_value . Вызывает аналогичную функцию для уровней (void Level::AddPoint(double x_value , Level *level)). Используется классом TalysCalculation для построения графиков зависимости результатов вычислений от переменных, задаваемых пользователем.