

48/14-00

ISSN 0234-5366



КРАТКИЕ СООБЩЕНИЯ ОИЯИ

JINR RAPID COMMUNICATIONS

5,6[97]-99

**Yields of the Rare-Earth
Neutron-Deficient Isotopes
in the Reactions of Mo Isotopes
with ^{40}Ca Ions**

☐

**Наблюдение медленных компонентов
солитоноподобной волновой структуры
в массивном образце меди
при возбуждении электронным пучком**

☐

**Разработка и исследование
многослойных дрейфовых камер (MDC-2)
с малым количеством вещества
для внутренней части спектрометра HADES**

☐

**Измерение температуры уранового образца,
облучаемого вторичными нейтронами**

☐

**Краевые эффекты в пропорциональной камере.
Влияние диэлектрической рамки**

☐

**An Object-Oriented Framework for the Hadronic
Monte-Carlo Event Generators**

☐

**Уран-238 как источник получения энергии
электроядерным методом**

☐

Издательский отдел ОИЯИ

ДУБНА

JINR Publishing Department

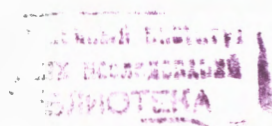
DUBNA

Объединенный институт ядерных исследований
Joint Institute for Nuclear Research

✓ 5,6[97]-99

КРАТКИЕ СООБЩЕНИЯ ОИЯИ
JINR RAPID COMMUNICATIONS

Дубна 1999



ОГЛАВЛЕНИЕ CONTENTS

J.Adam, P.Chaloun, Yu.P.Gangrsky, M.P.Ivanov, B.N.Markov, G.V.Mishinsky, Yu.E.Penionzhkevich, V.I.Stegailov, V.M.Tsupko-Sitnikov, A.B.Yakushev, V.I.Zhemenik, Xu Shuwei, Li Zhankui, Ma Ruichang, Zhang Sheng, Zhang Tianmei, Du Jinzhou, P.Zuzaan Yields of the Rare-Earth Neutron-Deficient Isotopes in the Reactions of Mo Isotopes with ^{40}Ca Ions И.Адам, П.Чалоун, Ю.П.Гангрский, М.П.Иванов, Б.Н.Марков, Г.В.Мышинский, Ю.Э.Пенионжкевич, В.И.Стегайлов, В.М.Цупко-Ситников, А.Б.Якушев, В.И.Жеменик, Сюй Шувей, Ли Чжанкуи, Ма Руичанг, Чжан Чен, Чжан Тианмей, Ду Чжинжоу, П.Зузаан Выходы нейтронодефицитных изотопов редкоземельных элементов в реакциях изотопов Мо с ионами ^{40}Ca	5
E.M.Kudryavtsev, A.V.Varava, A.V.Dedov, A.T.Komov Наблюдение медленных компонентов солитоноподобной волновой структуры в массивном образце меди при возбуждении электронным пучком E.M.Kudriavtsev, A.V.Varava, A.V.Dedov, A.T.Komov Observation of Slow Components of Solitonic-Type Wave Structure Excited by e-Beam in Massive Copper Sample	13
S.P.Chernenko, G.N.Agakishiev, V.F.Chepurnov, O.V.Fateev, L.N.Glonti, A.G.Petrov, V.N.Pechenov, L.P.Smykov, Yu.V.Zanevsky, H.Bokemeyer, H.Daues, J.Hegner, W.Koenig, H.Stelzer, C.Garabatos, J.Kempter, C.Muentz, J.Stroth, J.Wuestenfeld, P.Zumbruch Разработка и исследование многослойных дрейфовых камер (MDC-2) с малым количеством вещества для внутренней части спектрометра HADES S.P.Chernenko, G.N.Agakishiev, V.F.Chepurnov, O.V.Fateev, L.N.Glonti, A.G.Petrov, V.N.Pechenov, L.P.Smykov, Yu.V.Zanevsky, H.Bokemeyer, H.Daues, J.Hegner, W.Koenig, H.Stelzer, C.Garabatos, J.Kempter, C.Muentz, J.Stroth, J.Wuestenfeld, P.Zumbruch Development and Investigation of Low-Mass Multilayer Drift Chambers (MDC-2) for Inner Part of the HADES Spectrometer	22

В.И.Батин, Д.В.Батин, В.В.Борисов, Н.М.Владимирова, В.Г.Георгиев, В.М.Дробин, В.С.Королев, М.И.Кривопустов, С.В.Семашко, Л.С.Спасов, С.Г.Стеценко Измерение температуры уранового образца, облучаемого вторичными нейтронами V.I.Batin, D.V.Batin, V.V.Borisov, N.M.Vladimirova, V.G.Georgiev, V.M.Drobin, V.S.Korolev, M.I.Krivopustov, S.V.Semashko, L.S.Spassev, S.G.Stetsenko Temperature Measurement of the Uranium Sample Irradiated with Secondary Neutrons	33
Т.П.Топурия, М.Д.Шафранов Краевые эффекты в пропорциональной камере. Влияние диэлектрической рамки T.P.Topuria, M.D.Shafranov Edge Effects in Multiwire Proportional Chambers. The Influence of the Dielectric Frame	42
N.Amelin, M.Komogorov An Object-Oriented Framework for the Hadronic Monte-Carlo Event Generators Н.Амелин, М.Комогоров Объектно-ориентированный фреймворк для создания и использования генераторов событий взаимодействия частиц и ядер	52
В.П.Дмитриевский, Н.В.Сергеева Уран-238 как источник получения энергии электроядерным методом V.P.Dmitrievsky, N.V.Sergeeva Uranium-238 as a Source for Electronuclear Power Production	85

УДК 539.172

YIELDS OF THE RARE-EARTH NEUTRON-DEFICIENT ISOTOPES IN THE REACTIONS OF Mo ISOTOPES WITH ^{40}Ca IONS

J.Adam¹, P.Chaloun¹, Yu.P.Gangrsky, M.P.Ivanov, B.N.Markov, G.V.Mishinsky, Yu.E.Penionzhkevich, V.I.Stegailov, V.M.Tsupko-Sitnikov, A.B.Yakushev, V.I.Zhemenik, Xu Shuwei², Li Zhankui², Ma Ruichang², Zhang Sheng², Zhang Tianmei², Du Jinzhou², P.Zuzaan³

The results of the joint Russian-Chinese experiment on the measurements of the reaction cross sections of the rare-earth neutron-deficient isotopes production and the study of their decay scheme are presented. The studied nuclides were obtained in the reactions $^{92}\text{Mo} + ^{40}\text{Ca}$ and $^{97}\text{Mo} + ^{40}\text{Ca}$ on the 4-meter cyclotron of the Flerov Laboratory of Nuclear Reactions, JINR. The recoil nuclei were stopped in the inert gas and transported by the gas flow to the detectors. The single and coincidence spectra of γ -, x-rays and delayed protons were measured. The enhanced yield of the reactions with the charge particle evaporation was observed.

The investigation has been performed at the Flerov Laboratory of Nuclear Reactions, JINR.

Выходы нейтронодефицитных изотопов редкоземельных элементов в реакциях изотопов Мо с ионами ^{40}Ca

И.Адам и др.

Представлены результаты совместного российско-китайского эксперимента по измерению сечений реакций и исследованию свойств нейтронодефицитных изотопов редкоземельных элементов вблизи границы нуклонной стабильности. Изучаемые нуклиды были получены в реакциях $^{92}\text{Mo} + ^{40}\text{Ca}$ и $^{97}\text{Mo} + ^{40}\text{Ca}$ на четырехметровом циклотроне Лаборатории ядерных реакций ОИЯИ. Выбитые из мишени продукты реакций тормозились в газе и переносились газовым потоком к детекторам радиоактивного излучения. Измерялись спектры γ - и x-лучей, а также запаздывающих протонов и их совпадения. Наблюдался повышенный выход продуктов реакций с испусканием заряженных частиц.

Работа выполнена в Лаборатории ядерных реакций им.Г.Н.Флерова ОИЯИ.

¹Permanent address: Nuclear Physics Institute, Řež, Czech Republic

²Institute of Modern Physics, Lanzhou, China

³State University, Ulan-Bator, Mongolia

INTRODUCTION

The heavy ion reactions are the most effective way to produce the nuclei on the boundary of the nucleon stability. These reactions are the important source of the information about the mechanism of the complicated nuclear system interactions, the peculiarities of the high excited nuclei decay and the properties of the obtained nuclides. The fusion reactions with the evaporation of some number of neutrons are used in these investigations. The largest yields of the studied nuclei are obtained, if the target and bombarding ion are the lightest neutron-deficient stable isotopes.

Such kind of the experiments is performed in the Russian-Chinese cooperation. The purpose of this work is the study of the mechanism of the reactions between the complicated nuclei and the properties of the very neutron-deficient nuclides. The objects of the investigations are the reactions between the isotopes ^{92}Mo and ^{97}Mo and ions ^{40}Ca in the energy region 4 - 5 MeV/nucleon. The compound nuclei ^{132}Sm and ^{137}Sm are formed with the excitation energy of 70 - 100 MeV. They are neutron-deficient nuclei: ^{132}Sm has 12 neutron less than the lightest stable isotope ^{144}Sm and only 4 neutron more than the last proton stable isotope ^{128}Sm . A lot of neutron-deficient isotopes of the rare-earth element is formed at the decay of these compound nuclei. In the paper, the results of the first stage of this work are presented: the yields and the cross sections of the nuclides formed in the reaction $^{97}\text{Mo} + ^{40}\text{Ca}$. These values allow one to refer about the competition of different reaction channels at the de-excitation of the high excited and very neutron-deficient nuclei. The decay scheme for the most produced in the reaction nuclides are known, therefore the measurements of γ -spectra of these nuclides allow one to determine their yields.

EXPERIMENTAL SET-UP

The experiments were performed on the heavy ion 4-meter isochronous cyclotron of the Flerov Laboratory of Nuclear Reactions, JINR. The experimental method was based on the transportation of the reaction products by the gas flow to the detectors of the radioactive radiation.

The block-scheme of the experimental set-up is shown on Fig.1. The irradiated target was placed inside the chamber. This chamber was isolated from cyclotron vacuum by the Al-window (the thickness is 20 μm). The used Mo-targets were enriched by the isotopes ^{92}Mo and ^{97}Mo up to 95%. Their thickness was $\sim 2 \text{ mg/cm}^2$ and the loss of ^{40}Ca energy was $\sim 15 \text{ MeV}$. This energy loss was compared with the width of the excitation function for the production of the studied nuclides.

The intensive beam of $^{40}\text{Ca}^{+5}$ ions was produced in the ECR-ion source, placed outside the cyclotron [1]. The ^{40}Ca ions were accelerated up to the energy of 285 MeV and extracted from the cyclotron using a charge change by the thin C-foil. The energy of the ^{40}Ca ions was measured and controlled during the experiment. The thin scattering Au-foil and Si-detector placed before the target chamber were used for this purpose. The ^{40}Ca energy on the target was $\sim 200 \text{ MeV}$, it is the optimum energy for obtaining the studied nuclides. The intensity of ^{40}Ca beam in the target chamber was 10^{12} 1/s and does not change during the experiment.

The inert gas He flowed through the target chamber. The nuclei recoils are stopped in the gas and adsorbed by the aerosols. These aerosols were NaCl microcrystals produced in

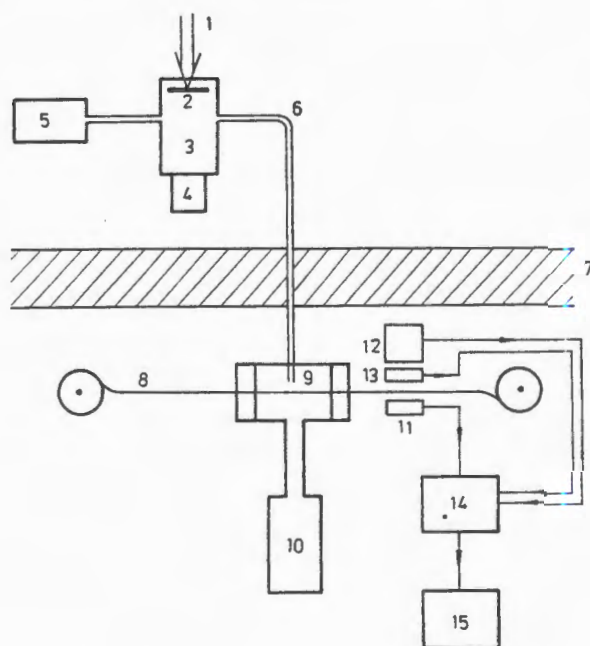


Fig. 1. Block-scheme of the experimental set-up: 1 — ion beam, 2 — target, 3 — gas chamber, 4 — Faraday cup, 5 — aerosol generator, 6 — capillar, 7 — shielding wall, 8 — moving tape, 9 — pumped chamber, 10 — pump, 11 — HpGe-detector for x-rays, 12 — HpGe-detector for γ -rays, 13 — Si-detector for the delayed protons, 14 — electronic blocks, 15 — PC

the special generator [2]. The pressure of He inside the chamber was ~ 1 atm. It was enough to stop the recoils, since their energy was less than 60 MeV and the range in He — less than 10 cm.

The gas flow through the chamber was produced by the pump of the 30 l/s productivity. The aerosols with the adsorbed recoils were drawn in the capillar by the flow and carried through the thick wall in the measuring room. The length of the capillar was 3 m, its diameter — 1 mm and time of the recoils transportation — 0.1 s. The nuclei recoils were collected on the moving tape inside the chamber. The pumping of the inert gas was realized through this chamber. The moving of the tape was stepwise with the stopping in front of the detectors. Three detectors were used for measurements of the nuclear radiation spectra: HpGe-detectors (100 cm³) for γ -radiation, thin HpGe-detector (10 mm) for x-rays and thin Si-detector for the delayed protons. Usually the period of tape stopping near the detectors was 20 s and the single and coincidence spectra as a function of time were measured and accumulated in PC.

EXPERIMENTAL RESULTS

The single x-ray spectrum of the $^{97}\text{Mo} + ^{40}\text{Ca}$ reaction products is shown in Fig.2. The energies of the observed lines correspond to K_{α} - and K_{β} -lines of the elements in the range $Z = 56 - 60$. It is obvious, that the origin of x-rays is the K-capture of the neutron-deficient

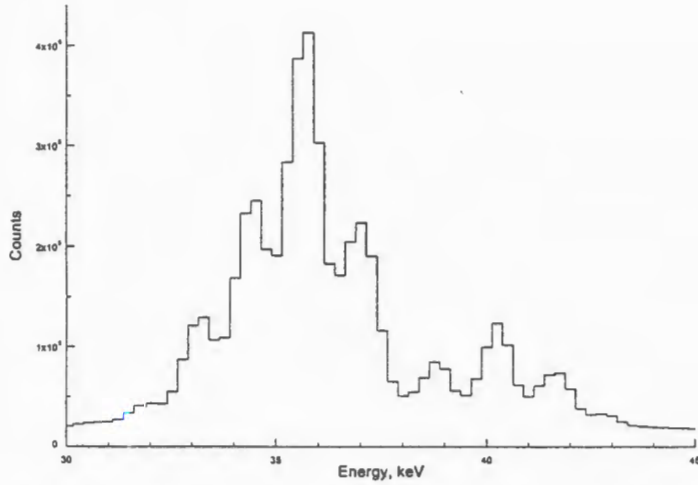


Fig. 2. Spectrum of the x-rays of the $^{97}\text{Mo} + ^{40}\text{Ca}$ reaction products

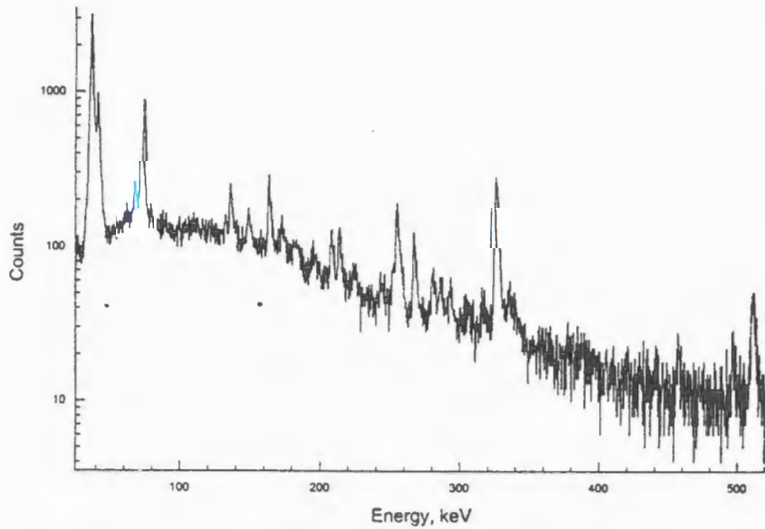


Fig. 3. Spectrum of the γ -radiation of the $^{97}\text{Mo} + ^{40}\text{Ca}$ reaction products

nuclides. Therefore the x-ray spectrum allows one to refer to the total element yield of the reaction products. It is seen from Fig.2, that the most intensive lines in the x-ray spectrum are the K_{α} - and K_{β} -lines of Ce, Pr, and Nd appearing after K-capture of Pr, Nd, and Pm, respectively. The weak K_{α} -lines after K-capture of Ce and La are observed also, but the yield of Pm x-ray is masked by the intensive line of Ce. The intensity of the K-capture branch in the decay of the produced nuclides is unknown as a rule. Therefore it is possible to get only some estimates of total yields for different elements produced in the reaction $^{97}\text{Mo} + ^{40}\text{Ca}$. It is seen that the most yields have the isotopes of Pr, Nd, and Pm obtained at the evaporation from the compound nuclei of 1-3 protons of ^{137}Sm and some number of neutrons.

Table 1. The observed γ -lines, their relative intensity and identification

E_γ , keV	I_γ , %	$T_{1/2}$	Nuclide
56.0	8	5.1 m	^{127}La
68.2	11	3.5 m	^{129}Ce
87.1	12	27 s	^{131}Nd
120.0	6	31 s	^{127}Ce
131.0	9	25 m	^{130}Ce
147.8	7	5.1 m	^{128}Ce
230.3	12	5.0 m	^{131}Ce
253.8	100	40 s	^{130}Pr
255.9	21	60 s	^{126}La
266.0	25	5.7 s	^{131}Pr
279.1	11	11.6 m	^{129}La
284.0	66	4.9 m	^{128}La
325.3	57	1.6 m	^{132}Pr
357.4	5	8.7 m	^{130}La
399.5	20	3.1 s	^{128}Pr
567.0	4	88 s	^{132}Nd

Table 2. Relative yields and cross sections of the products of the reaction $^{97}\text{Mo} + ^{40}\text{Ca}$

Nuclei	Evaporated particles	Y_{rel} , %	σ , mb
^{132}Sm	5n	<2.5	<0.2
^{133}Sm	4n	<2.5	<0.2
^{132}Pm	p4n	<1.2	<0.1
^{133}Pm	p3n	<1.2	<0.1
^{132}Nd	2p3n(α n)	28	2.2
^{131}Nd	2p4n(α 2n)	35	2.8
^{132}Pr	3p2n(α p)	39	3.1
^{131}Pr	3p3n(α pn)	45	3.5
^{130}Pr	3p4n(α p2n)	100	8.0
^{131}Ce	4p2n(α 2p)	22	1.8
^{130}Ce	4p3n(α 2pn)	13	1.1
^{129}Ce	4p4n(2 α)	7.5	0.6
^{130}La	5p2n(α 3p)	20	1.6
^{129}La	5p3n(α 3pn)	47	3.7
^{128}La	5p4n(2 α p)	17	1.4
^{127}La	5p5n(2 α pn)	10	0.8
^{126}La	5p6n(2 α p2n)	6.2	0.5

The single γ -ray spectra are very complicated, but the spectra of γ -x-coincidence are more simple (one example of these spectra is shown in Fig.3). The analysis of these spectra allows one to identify a lot of γ -lines and attribute them to the definite nuclides. The recent table of isotopes [3] was used for this identification. The identified γ -lines, their intensities (relative to the most intensive line 253.8 keV of ^{130}Pr), the obtained nuclides and their half-lives are presented in Table 1.

For the majority of the identified nuclides, the decay scheme and the absolute intensity of the observed γ -lines are known. It is possible in these cases to determine the yields and the cross sections of the nuclei obtained in the studied reaction $^{97}\text{Mo} + ^{40}\text{Ca}$. Some nuclides

are formed not only at the proton and neutron evaporation from the compound nuclei, but after β^+ -decay of the nuclei with the same mass number. In these cases the corrections on the population of these nuclei in the β^+ -chains are induced. The identified nuclides, the evaporated in the reactions particles, the yields and the cross sections are presented in Table 2. The sum of these cross sections coincides with the total cross section calculated using ALICE code [4] for the fusion reaction $^{97}\text{Mo} + ^{40}\text{Ca}$.

DISCUSSION OF RESULTS

In Fig.4 is presented the chapter of isotopes with the obtained nuclides. It is seen that the obtained nuclides are situated in the range $Z = 57 - 60$ (La - Nd) and $A = 126 - 132$. There were not observed the reaction products after the evaporation of only neutron (Sm isotopes) and only protons (^{132}La , ^{133}Ce). The largest yields have the nuclei formed at the evaporation near the same number of protons and neutrons. The lightest nuclides (isotopes Ce and La) could be formed only at the emission of the α -particles instead of the single neutrons and protons.

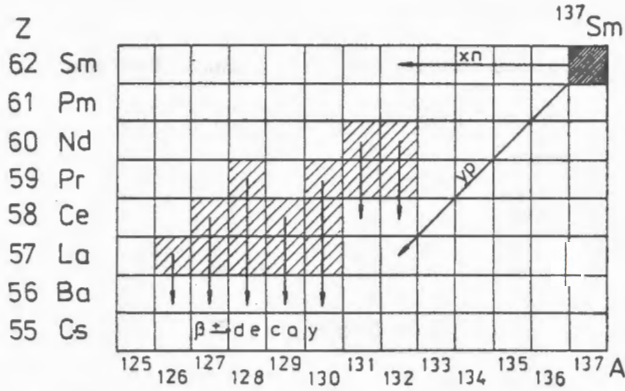


Fig. 4. Part of nuclide chapter in the region of the produced nuclei

According to the statistical model [5,6] the probability of the evaporation of any particle from the compound nuclei is expressed by the relation:

$$\Gamma_i = \hbar \int_0^{E-B_i} W_i(\varepsilon) d\varepsilon, \quad (1)$$

where E is the excitation energy of the compound nuclei; B_i and ε_i , the binding and kinetic energies of the evaporated particle

$$W_i(\varepsilon) \sim \sigma(E, \varepsilon) \varepsilon \rho(E - B_i - \varepsilon), \quad (2)$$

where $\sigma(E, \varepsilon)$ is the cross section of the back reaction; $\rho(E - B_i - \varepsilon)$, the level density of the nucleus after the evaporation of the particle (proton or neutron). In the relation (2) the most important is the value $\rho(E - B_i - \varepsilon)$, since its energy dependence is exponential. Therefore the competition between proton and neutron evaporation depends mostly on the value of residual

nucleus level density $\rho(E - B_i - \epsilon)$. This value is determined by the excitation energy of the residual nucleus and, consequently, the energy, removed by the evaporated particle $\epsilon + B_i$. In the case of the neutrons $\epsilon \sim 2T$ (T is the temperature of the compound nuclei), but in the case of protons $\epsilon \sim 2T + B_c$ (B_c — coulomb barrier for the protons). Therefore the evaporation of the neutron from the compound nucleus is more probable for the nuclei with nearly the same values of proton and neutron binding energy (they are the nuclei close to the β -stability valley). But in neutron-deficient nuclei $B_p < B_n$ and evaporation protons and neutrons lead to nearly the same excitation energy of the residual nucleus. In this case the probability of the proton and neutron evaporation became near equal.

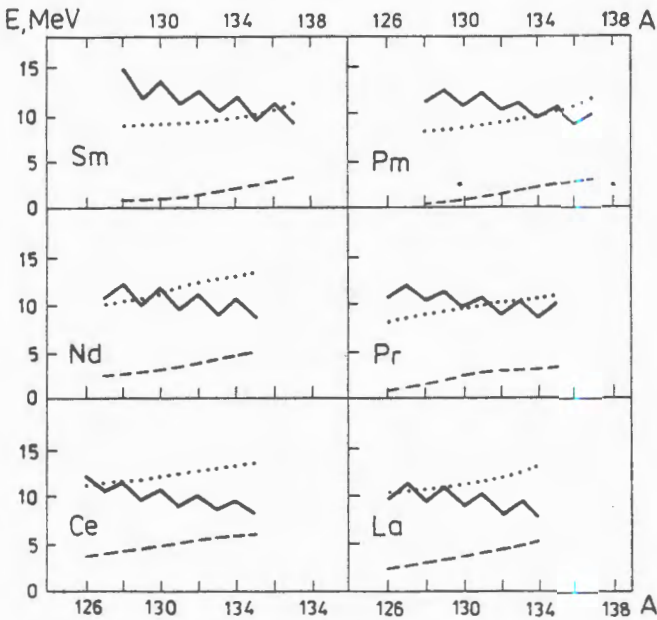


Fig. 5. Binding energies of neutrons (-) and protons (- -) including their coulomb barrier (...)

In Fig.5 are shown the binding energies of the proton and neutron in the observed isotope (La–Sm), including the coulomb barrier for the protons. The experimental values of the coulomb barrier were used [7]. It is seen that in the most Sm and Pm isotopes the removed energy is more for the neutrons. In Ce and La isotopes the opposite picture is observed. Therefore in Sm and Pm isotopes the proton evaporation is preferable, and it explains their very low yields in the reaction $^{97}\text{Mo} + ^{40}\text{Ca} \rightarrow ^{137}\text{Sm}$. The obtained in the reaction nuclei are well deformed. There is the indication on the lowering of the coulomb barrier for such nuclei [8]. This phenomenon increases the probability of proton emission.

The comparison of the experimental and calculated cross sections or yield ratios of the isotopes with the same mass number A allows one to obtain the new information about the level density of the very neutron-deficient nuclei close to proton drip line. This information will help to choose the optimal conditions for the production of new isotopes in this region of Z and A . It is expected that the fusion reactions $^{92}\text{Mo} + ^{40}\text{Ca}$ and $^{196}\text{Ru} + ^{32}\text{S}$ are the most perspective for this purpose.

The authors thank Prof. Yu. Oganessian, Prof. M. Itkis and Prof. V. Kalinnikov for the attention to this work, Dr. E. Cherepanov for the fruitful discussions, the crew of 4-meter cyclotron FLNR for the providing a stable and intensive beam of ^{40}Ca . The work was supported by the Russian Federal Found Investigations (RFFI) and Chinese Scientific Found (grant No.96-02-00036).

References

1. Gikal B.N., Gulbekian G.G., Kutner V.B. — In: Proc. of the 15th Int. Conf. on Cyclotron and Their Applications, Caen, France, 1998.
2. Zvara I., Domanov Yu.P., Shalaevsky M.R., Petrov D.V. — JINR, 12-80-48, Dubna, 1980.
3. Browne E., Firestone A.B. — Table of Radioactive Isotopes, Ed. V.S. Shirley, N.Y., 1986.
4. Reisdorf W. — Z. Phys. A, 1981, v.300, p.227.
5. Weisskopf V. — Phys. Rev., 1937, v.52, p.295.
6. Gilbert A., Cameron A.C.W. — Canad. J. Phys., 1965, v.43, p.1446.
7. Vaz L.C., Alexander J.M. — Z. Phys. A, 1984, v.318, p.231.
8. Davids C.N., Woods P.J., Seweryniak D. et.al. — Phys. Rev. Lett., 1998, v.80, p.1849.

УДК 536.2.01+621.039.63+537.533

НАБЛЮДЕНИЕ МЕДЛЕННЫХ КОМПОНЕНТОВ СОЛИТОНОПОДОБНОЙ ВОЛНОВОЙ СТРУКТУРЫ В МАССИВНОМ ОБРАЗЦЕ МЕДИ ПРИ ВОЗБУЖДЕНИИ ЭЛЕКТРОННЫМ ПУЧКОМ

Е.М.Кудрявцев^{1,2}, А.В.Варава³, А.В.Дедов³, А.Т.Комов³

Термопарные записи колебаний температуры стенки охлаждаемой медной трубки толщиной 3 мм, на которую подан электронный пучок с плотностью 0,4–7 кВт/см², обнаруживают несколько медленных компонентов солитоноподобной волновой структуры. Они зарождаются на внешней и внутренней стенках облучаемого участка трубки и распространяются в нем со скоростями порядка 10⁻² см/с, многократно отражаясь от стенок. Скорости последующих компонентов отличаются приблизительно вдвое. Ранее аналогичные волновые структуры наблюдались для различных материалов при облучении их ИК-лазерными импульсами с плотностью мощности порядка 10 кВт/см².

Работа выполнена в ФИАН им. П.Н.Лебедева и МЭИ(ТУ) при частичной поддержке РФФИ, проект 97-02-17954-а.

Observation of Slow Components of Solitonic-Type Wave Structure Excited by e-Beam in Massive Copper Sample

E.M.Kudriavtsev et al.

There were shown several slow components of solitonic-type wave structure in the thermocouple temperature variation records of the cooled copper tube 3 mm thick. The tube was irradiated by electron beam with density (0.4–7.0) kW/cm². These components arise on the external and internal surfaces of irradiated tube area and spread within it with the velocities of the order of 10⁻² cm/sec being repeatedly reflected from the walls. The subsequent component velocities differ approximately twice. The analogous wave structures were observed previously for the different materials at its IR-laser pulse irradiation with the density of the order of 10 kW/cm².

The investigation has been performed at the Lebedev Physical Institute of RAS (Moscow) and Moscow Power Institute (Technical University) and partially supported by RFBR, project 97-02-17954-a.

¹Физический институт им. П.Н.Лебедева РАН, Отделение оптики, Москва

²E-mail address: kudri@sgi.lpi.msk.su or <kudriavt@sci.lpi.msk.su>

³Московский энергетический институт (технический университет), кафедра общей физики и ядерного синтеза

1. ВВЕДЕНИЕ

В серии работ, посвященных исследованию воздействия импульсного лазерного ИК-излучения сначала на ВТСП-материалы [1,2], а затем и на образцы четырех главных типов кристаллов и ряда аморфных тел (см. обзор [3]), было обнаружено новое явление, названное волной изменения отражения и проводимости (ВИОП) по первым методам регистрации волны. Она представляет собой трехмерную солитоноподобную волновую структуру, в которой из области генерации в ответ на один возбуждающий импульс достаточной плотности (около 10 кВт/см²) по поверхности материала и сквозь него расходится целая последовательность одиночных волн — компонентов ВИОП. Они имеют одинаковый и постоянный знак амплитуды (положительный или отрицательный), а величина амплитуды слабо падает с расстоянием. Каждый компонент ВИОП имеет постоянную скорость и отражается от стенок образца с малыми потерями энергии. Скорости последующих компонентов U_i каждый раз уменьшаются примерно вдвое, начиная от продольной скорости звука v_i (порядка км/с) до скоростей порядка мм/с, согласно выражению

$$U_i = (2)^{-i} v_i, \quad (1)$$

где индекс i , как установлено в экспериментах (см. [3]), может принимать значения от 0 до 25. При этом не видно ограничений на его дальнейшее увеличение.

ВИОП была зарегистрирована сначала по обратимому изменению отражения для видимого света по методике наблюдения изменений отражения в геометрии Брюстера, а затем по изменению проводимости в образце под действием этой волны. Затем была предложена и реализована методика регистрации изменений давления как третьего независимо измеряемого параметра при прохождении ВИОП. В дальнейшем для ее регистрации использовалось изменение пропускания образцов в видимой или ИК-областях спектра и тепловое излучение образца (с помощью термокамеры) [3]. Измерения колебаний температуры с помощью такого надежного средства, как термopара, представленные в настоящей работе (см. также [4]), делают результаты исследований ВИОП особенно достоверными.

Вопрос о возможной роли солитонов в переносе тепловой энергии в твердых телах обсуждался уже давно, в основном теоретически (см. краткий обзор в [3]). М.Тода [5] пришел к заключению, что солитоны играют важную роль в теплопроводности нелинейных кристаллических решеток с нарушениями. С экспериментальной точки зрения вопрос обсуждался в [6].

В течение ряда лет на кафедре общей физики и ядерного синтеза Московского энергетического института (технического университета) исследовались возможности прерывания корпускулярных потоков большой мощности с помощью экранов, охлаждаемых водой, протекающей под высоким давлением через экран. Быстрый нагрев стенки экрана до температур, достигающих 500 — 600° С, приводит к различным режимам теплоотвода, что может выражаться в наблюдаемых в разных участках экрана колебаниях температуры.

Как оказалось, по крайней мере часть регистрируемых колебаний температур вызвана другой причиной — с возбуждением указанных выше медленных солитоноподобных компонентов ВИОП, что и продемонстрировано в настоящей статье.

2. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Подробное описание экспериментальной установки приведено в [7,8]. Она позволяет моделировать процессы теплообмена при вынужденной конвекции теплоносителя (воды) и одностороннем корпускулярном нагреве рабочего участка экрана с помощью пучка электронов от электронной пушки ЭЛА 60/15-Т.

Рабочим участком служила поверхность 10×10 мм² медной трубы наружного диаметра 14 мм и внутреннего диаметра 8 мм, через которую прокачивалась холодная вода под высоким давлением. Труба была помещена в вакуумную камеру с форвакуумной и диффузионной откачкой (рабочее давление $(2-3) \cdot 10^{-3}$ Па). Строчно-кадровая развертка электронного пучка обеспечивала сканирование поверхности нагрева с частотой кадра 10 кГц при количестве строк в кадре 64. Этим обеспечивалась однородность распределения мощности по сечению пучка. Сама мощность регулировалась изменением анодного тока. В описываемой серии опытов она составляла величины 0,4 – 7 кВт/см².

Выведение пучка на полную мощность происходило за время ~ 1 с. Регистрация температуры (с ошибкой $\sim 1\%$) производилась хромель-копелевой термопарой с инерционностью порядка мс, вмонтированной в рабочий участок трубы со стороны внутренней поверхности на глубину $\sim 0,5$ мм.

Система измерений, сбора и обработки информации для большого числа необходимых параметров, опускаемых нами здесь (см. подробнее [7]), была полностью автоматизирована в стандарте приборного интерфейса МЭК 625.1. Измерения температуры по термоЭДС указанной термопары производились с помощью универсального цифрового вольтметра В7-43, обеспечивающего частоту измерений и записи 7 Гц.

3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для настоящей работы были отобраны 8 экспериментов, температурные записи которых представлены на рис. 1. Во всей серии температура воды на входе в измерительную систему была 10° С, а среднее давление воды 12,6 атм.

В каждом эксперименте задавались свои условия расхода теплоносителя. После включения электронного пучка требуемой мощности происходил нагрев образца. За времена около 10 – 20 с температура T внутренней стенки облучаемой медной трубки достигала значений, близких к максимальным, и далее менялась уже незначительно.

3.1. Режим нагрева с кипением. На рис. 2а представлена в достаточно крупном масштабе сглаженная по 5 точкам часть одной из таких записей (запись 1 на рис. 1; опыт № 1, см. ниже таблицу).

Эта запись (как и остальные измерения) соответствует вариациям температуры медной трубки на расстоянии 0,5 мм от ее внутренней поверхности и, соответственно, на расстоянии 2,5 мм от облучаемой электронным пучком поверхности.

На записи, на фоне медленного роста T , видны колебания температуры ΔT , достигающие 3 – 5° С, или 0,6 – 1,0% от всего изменения температуры $T_{\text{макс}} - T_0$. Первый максимум A задержан относительно включения пучка (момент $t = 0$) на ~ 17 с.

На рис. 2б представлена x - t -диаграмма, характеризующая волновой процесс, происходящий в рассматриваемом образце меди после включения электронного пучка, который приводит к изменениям температуры, отмеченным на рис. 2а.

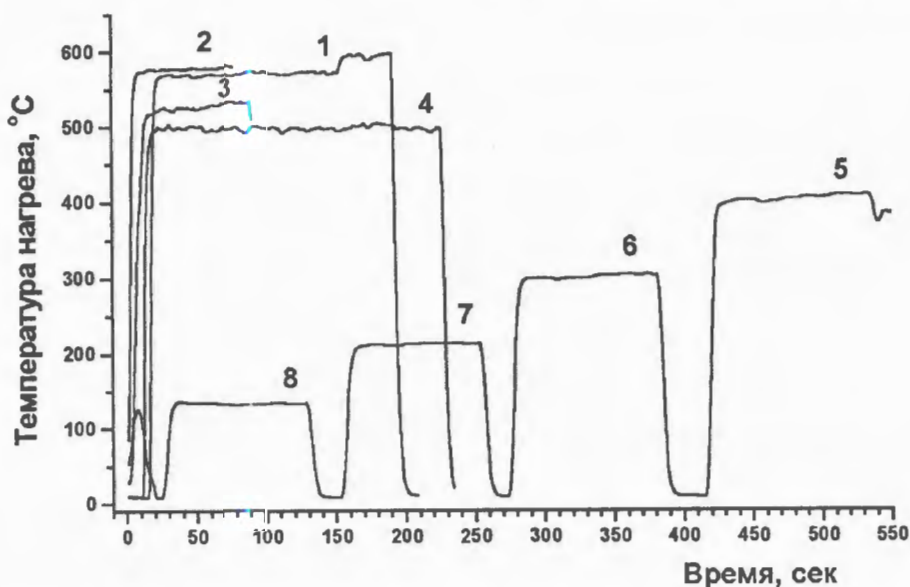


Рис. 1. Общий вид температурных записей экспериментов, отобранных для обработки

При этом мы исходили из предположения, что при прохождении ВЮП через область расположения термопары последняя регистрирует *увеличение* температуры. Вообще говоря, можно построить аналогичную диаграмму, исходя из предположения, что приход ВЮП в конкретном случае меди вызывает *снижение* температуры, что не противоречит свойствам этой солитоноподобной волновой структуры согласно нашим исследованиям [3]. Такая диаграмма хуже согласуется с экспериментальной температурной записью рис. 2а, но достаточно логична. В связи с тем, что адекватный механизм ВЮП еще не разработан, вопрос о *знаке* изменения ΔT в этой волне, идущей через медь, требует дальнейшей экспериментальной и теоретической проработки, однако главное утверждение о регистрации компонента ВЮП в описываемых опытах остается в силе.

На рис. 2б штриховыми отрезками представлено движение с приблизительно постоянной скоростью одиночной волны, образовавшейся на лицевой, облучаемой пучком поверхности. Она движется сперва к внутренней поверхности медной трубы, которой достигает, приблизительно, через 17 с (это время соответствует первому максимуму А на кривой рис. 2а. Затем волна отражается от этой внутренней поверхности и движется с той же (или несколько меньшей) скоростью к лицевой поверхности, где снова отражается, обуславливая максимум В при подходе к термопаре (положению термопары на $x-t$ -диаграмме рис. 2б соответствует горизонтальная штрихпунктирная прямая при $x = 2,5$ мм). Далее процесс повторяется многократно. Каждый раз, когда волна проходит область расположения термопары, она вызывает подъем средней температуры на указанные величины порядка 1% (пики А, Б, В, Г).

Однако пики А, Б, В, Г на кривой рис. 2а являются составляющими дублетов, в которые, кроме них, входят несколько меньшие по интенсивности пики a_1, b_1, v_1, g_1 . Их регулярность позволяет предположить, что они, как и А, Б, В, Г, обусловлены приходом к той же термопаре другой одиночной волны, несколько меньшей по интенсивности,

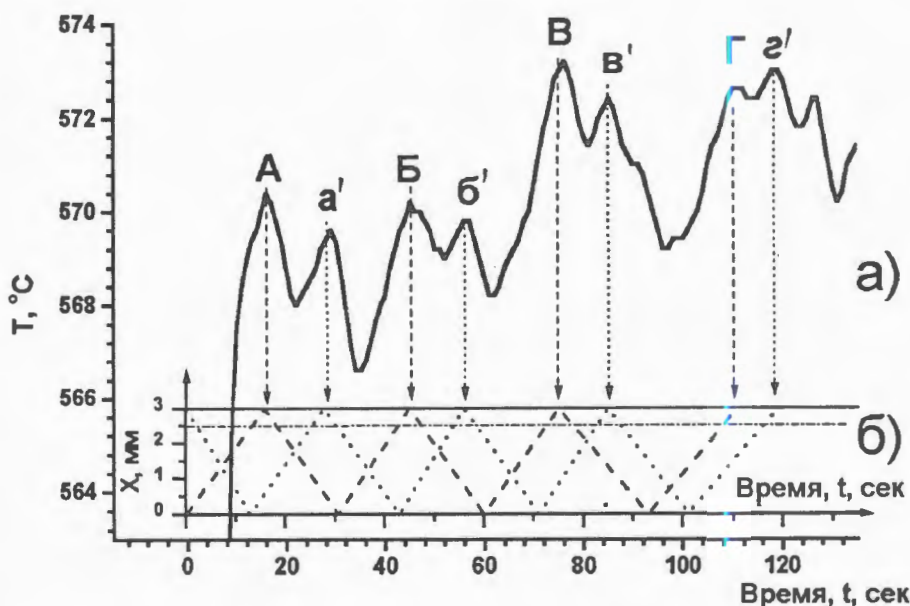


Рис. 2. Случай режима нагрева с кипением. а) Вариации температуры; б) соответствующая им x - t -диаграмма волнового процесса

движущейся приблизительно с той же скоростью, но зародившейся на внутренней поверхности медной трубы (пунктирная линия на рис. 2б). Из расположения пиков a' , $б'$, $в'$, $г'$ можно заключить, что постоянная скорость обуславливающей их волны несколько уменьшается после каждого отражения, но это ослабление менее выражено, чем для волны, обуславливающей пики А, Б, В, Г.

Здесь уместно отметить, что на рис. 2а третий дублет $Вв'$ и, особенно, четвертый $Гг'$ выглядят уже как триплеты, но этим усложнением структуры и вопросом, с чем связаны третьи компоненты таких триплетов, мы пока пренебрегаем.

Зная расстояние, проходимое каждой одиночной волной, и замерив последующие моменты прихода этих волн к термопаре, легко рассчитать их скорости. При этом, по нашим оценкам, ошибка в измерении расстояния мала, так как при периодическом повторении процесса достаточно знать толщину образца, что можно сделать с большой точностью. Ошибка в измерении моментов времени прихода волны к приемнику — термопаре может оцениваться величиной $\sim 10\%$, что и определяет ошибку в измерении скорости ВАОП в данных экспериментах.

Результаты такого расчета скоростей приведены в таблице для всей серии из 8 экспериментов. Опыту № 1, результаты которого открывают таблицу, соответствует запись температуры, представленная на рис. 2а, и помогающая ее расшифровать x - t -диаграмма рис. 2б.

В соответствии с выражением (1), обе найденные для опыта № 1 скорости соответствуют 25-му компоненту ВАОП, превышая его на $\sim 40\%$ (в расчете использовано табличное значение продольной скорости звука для меди, равное $4,70 \cdot 10^5$ см/с из [9]). Это расхождение частично можно отнести за счет распространения ВАОП по металлу,

Таблица. Условия и результаты экспериментов

№№ опы- тов	Время до пер- вого макси- мума, с	Темпера- тура нагре- ва, ° С	Длитель- ность разверт- ки, с	Скорость ВИОП (от лицевой поверхнос- ти), см/с	Скорость ВИОП (от внутренней поверхнос- ти), см/с	Номер компо- нента ВИОП	Отклоне- ние от рассчитан- ной по (1) скорости, %
1	17	570	135	0,0195	0,0198	U_{25}	+40
2	18	578	80	0,0208	0,0173	U_{25}	+36
				0,0416	—	U_{24}	+49
3	25	525	90	0,0162	0,0173	U_{25}	+20
4	21	497	210	0,0132	0,0139	U_{25}	—3, 2
5	27	405	120	0,010	0,010	U_{26}	+43
6	15	300	110	0,0190	0,0213	U_{25}	+44
7	22	212	100	0,011	0,011	U_{26}	+57
8	10	137	100	0,0182	0,0182	U_{25}	+30

нагретому довольно значительно по отношению к его $T_{\text{плвл.}} = 1629^\circ \text{С}$. Увеличение скорости ВИОП наблюдалось ранее в экспериментах с диэлектриками, когда повышалась температура образца [3].

В первых четырех строках таблицы даны результаты экспериментов, проведенных при близких условиях малого расхода охлаждающей воды и высоких мощностей в пучке (нагрев медной трубки достигал $\sim 500 - 580^\circ \text{С}$). Для всех этих опытов были построены $x-t$ -диаграммы, которые достаточно хорошо повторяют главные особенности рис. 2б. Все данные этой первой части таблицы с некоторым разбросом согласуются с рассчитанной по (1) скорости 25-го компонента ВИОП (компонента с индексом $i = 25$).

На температурных записях в опытах № 3 и № 4, которые мы не воспроизводим здесь, кроме тех же пиков, отождествленных как U_{25} , в промежутках между ними видны небольшие максимумы, соответствующие проявлениям предыдущего компонента ВИОП, то есть U_{24} , образовавшейся на лицевой стороне. Аналогичный компонент, образовавшийся на внутренней стороне, зарегистрировать не представляется возможным. Скорости этих компонентов U_{24} на записях опытов № 3 и № 4 приблизительно вдвое больше величины U_{25} (оценивать ошибку в величине отношения $U_{24}/U_{25} = 2$ из-за недостатка данных нецелесообразно).

Возбуждение ВИОП электронным пучком позволило получить новую информацию об условиях возбуждения этих волн по сравнению с возбуждением лазерными импульсами [3]. Это относится к образованию волн *передним фронтом возбуждающего импульса*, поскольку в описываемых опытах возбуждение компонентов происходило, практически, только в начале эксперимента, когда мощность пучка менялась, достигая постоянного уровня (если опустить неясный пока вопрос о возникновении третьих компонентов дублетов на рис. 2а).

Кроме того, важным для разработки механизма явления результатом исследований ВИОП с возбуждением электронным пучком является образование некоторого i -го компонента волны на противоположной поверхности образца, *непрозрачного для частиц пучка*, одновременно с аналогичным i -м компонентом волны на облучаемой поверхности. Дело в том, что образование некоторого компонента волны на противоположной поверхности образца является обычным в лазерных опытах с диэлектриками, прозрач-

ными для ИК-излучения, возбуждающего ВАОП [3], но для непрозрачных объектов таких результатов до сих пор не было зафиксировано.

3.2. Режим нагрева без кипения. Как уже отмечалось во Введении, одной из предполагаемых причин, вызывающих колебания температуры, могут служить изменения в режиме теплопередачи. Появление колебаний может быть связано с наступлением сначала пузырькового режима кипения на внутренней (охлаждаемой) стенке трубки, который при дальнейшем увеличении температуры стенки переходит в режим плёночного кипения. Этот режим характеризуется нестабильностью существования паровой плёнки на внутренней стенке трубки и ее периодическими срывами.

В связи со сказанным были исследованы режимы нагрева электронным пучком медной трубы при таких больших расходах охлаждающей воды и таких малых мощностях пучка, когда кипение воды отсутствует.

В таблице этим условиям соответствует опыт № 8 и близко к ним подходит опыт № 7. Опыты №№ 5,6 соответствуют промежуточным условиям.

На рис. 3а,б представлены запись хода температуры и x - t -диаграмма волновых процессов для опыта № 8.

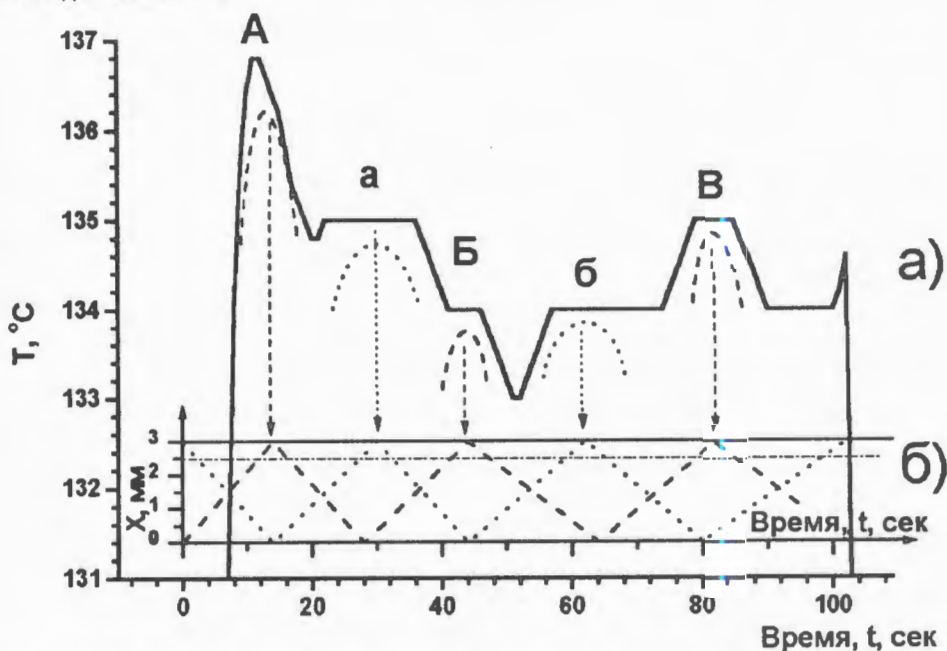


Рис. 3. Случай режима нагрева без кипения. а) Вариации температуры; б) соответствующая им x - t -диаграмма волнового процесса

Поскольку чувствительность схемы не менялась, а сигнал термопары при снижении температуры от 57 $^{\circ}\text{C}$ до 13 $^{\circ}\text{C}$ падает весьма существенно (в 4,2 раза), температурный ход на рис. 3а содержит гораздо меньше деталей, чем на рис. 2а, и ожидаемые пики ΔT разрешены хуже (их величина не превышает 1 – 1,5 $^{\circ}\text{C}$). Однако, исходя из тех же предположений, что и ранее, можно предложить аналогичную x - t -диаграмму процессов, рис. 3б). Кстати, на рис. 3б ширины пиков для двух волн явно различны (более широкие

пики — для волны, образованной на внутренней стороне). Это подтверждает справедливость расшифровки картины волнового процесса, представленного на $x-t$ -диаграмме, хотя изменения интенсивности последующих пиков А, Б, В на рис. 3а требуют дополнительного изучения.

Из приведенной таблицы следует, что для опытов № 8 и № 7, в которых процесс кипения можно не учитывать, измеренные скорости обеих одиночных волн (возникающих на лицевой и внутренней поверхностях медной трубки) такие же, как в остальных опытах, выполненных при более высоких температурах, когда кипением уже нельзя пренебречь (№№ 6,5) или когда процесс кипения можно считать определяющим для теплопередачи (№№ 1-4).

Следует отметить, что данные таблицы без дополнительных экспериментов не позволяют пока объяснить, какие условия в опытах с № 5 по № 8 приводят к тому, что в них проявляется то 25-й, то 26-й компоненты ВИОП.

4. ВЫВОДЫ

1. Результаты представленной серии из 8 опытов показывают, что при облучении электронным пучком с плотностью мощности 0,4 – 7 кВт/см² массивного образца меди в последнем возбуждаются медленные компоненты ВИОП, которые приблизительно соответствуют расчетным (по формуле (1)) значениям 24-го, 25-го и 26-го компонентов U_{24} , U_{25} , U_{26} этой волновой структуры.

Поскольку описываемые температурные записи были случайным образом отобраны из большого массива материалов (по наибольшей их длительности) и все они, без исключения, согласуются с $x-t$ -диаграммами волновых процессов, представленными на рис. 2а,б; 3а,б, наш вывод можно считать вполне достоверным.

2. Из сравнения результатов опытов, проведенных при существенно различных температурах (от 13° С до 58° С), можно видеть, что обнаруженные температурные колебания связаны, по крайней мере, не только с изменением режима теплопередачи между теплоносителем и исследуемым образцом меди. Тем самым подтверждается вывод, что ВИОП с ее компонентами играет определённую роль в процессе теплопереноса [6].

3. С точки зрения исследования ВИОП возбуждение их электронным пучком позволило получить новую информацию об условиях возбуждения этих волн (образование волн передним фронтом возбуждающего импульса; образование волны на противоположной поверхности образца, непрозрачного для частиц пучка, одновременно с волной на облучаемой поверхности). Эти данные необходимо учитывать при разработке механизма явления, который охватывал бы всю совокупность его свойств.

Авторы благодарны РФФИ за поддержку проекта 97-02-17954-а и XII Школы-семинара под руководством академика РАН А.И. Леонтьева, которая позволила осуществить настоящую работу. Авторы благодарны также С.Д.Зотову за помощь при обработке результатов и подготовке статьи.

Литература

1. Kudriavtsev E.M. et al. — Material Research Society - Meeting of FaU'92, Book of abstracts, H3, 25, Boston, USA, 1992.

2. Kudriavtsev E.M. et al. — Physica C, 1994, v.234-240, p.1439.
3. Kudriavtsev E.M. — In: High-Power Laser Ablation, Proceedings of SPIE, 1998, v.3343, p.411.
4. Кудрявцев Е.М. и др. — Краткие сообщения по физике ФИАН (принято к публикации в № 11 за 1999 год).
5. Toda M. — Physica Scripta, 1979, v.20, p.424.
6. Кудрявцев Е.М., Емельянов В.И., Утрик М. — В сб.: Проблемы газодинамики и теплообмена в энергетических установках (XII Школа-семинар молодых ученых и специалистов под руководством академика РАН А.И.Леонова, Москва, 25–28 мая 1999 г.). М.: изд-во МЭИ, 1999. с.298.
7. Дедов А.В., Комов А.Т. — там же, с.287.
8. Касаткин А.П., Комов А.Т., Наумов В.К. и др. — ВАНТ. Серия: Термоядерный синтез, 1993, вып.1–2, с.60.
9. Бабичев А.П. и др. — Физические величины. Справочник. М.: Энергоатомиздат, 1991.

УДК 539.1.074

РАЗРАБОТКА И ИССЛЕДОВАНИЕ МНОГОСЛОЙНЫХ ДРЕЙФОВЫХ КАМЕР (MDC-2) С МАЛЫМ КОЛИЧЕСТВОМ ВЕЩЕСТВА ДЛЯ ВНУТРЕННЕЙ ЧАСТИ СПЕКТРОМЕТРА HADES

С.П.Черненко, Г.Н.Агакишиев, В.Ф.Чепурнов, О.В.Фатеев, Л.Н.Глонти, А.Г.Петров, В.Н.Печенов, Л.П.Смыков, Ю.В.Заневский, Г.Бокемайер¹, Х.Дауес¹, Й.Хинэ¹, В.Кениг¹, Г.Штельцер¹, Ч.Гарабатос², И.Кемптер², К.Мюнц², И.Штрот², Е.Вустенфельд², П.Цумбрух²

HADES — широкоапертурный диэлектронный спектрометр, создаваемый в GSI (Дармштадт) для регистрации лептонных пар, образующихся в результате столкновений тяжелых ядер. HADES способен работать при интенсивностях пучка до 10^8 частиц/с. Детектор имеет геометрический аксептанс почти 40% для e^+e^- — пар и разрешение по массам 0,8% (σ) для ρ - и ω -мезонов. Четыре системы дрейфовых камер с малой массой предназначены для восстановления треков заряженных частиц. Относительная радиационная длина трековой системы не превышает 0,5% за счет использования газовой смеси на основе гелия и алюминиевой проволоки при создании катодных и анодных плоскостей. Разработан полномасштабный прототип дрейфовой камеры с малой массой для внутренней части спектрометра HADES. Прототип тестировался в лабораторных условиях с источниками ^{55}Fe и ^{90}Sr в ОИЯИ и GSI, а также на пучке синхротрона SIS (GSI). Пространственное разрешение ~ 70 мкм (σ) было получено на газовой смеси $\text{He} - \text{iC}_4\text{H}_{10}$ (60:40).

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Development and Investigation of Low-Mass Multilayer Drift Chambers (MDC-2) for Inner Part of the HADES Spectrometer

S.P.Chernenko et al.

HADES is a High Acceptance Di-Electron Spectrometer for measuring lepton pairs produced in heavy-ion induced reactions presently built up at GSI (Darmstadt). HADES is able to run at high intensities of up to 10^8 beam particles per second. The detector has a geometrical acceptance of almost 40% for e^+e^- pairs and a mass resolution of 0.8% (σ) for ρ and ω mesons. Four low-mass drift chamber systems serve for charged particle track reconstruction. The relative radiation length of the tracking system is kept

¹GSI, 64291, Дармштадт, Германия

²Университет Иоганна Вольфганга Гете, 60486, Франкфурт, Германия

below 0.5% by using helium based gas mixtures and Aluminium wires for the field wires and cathode planes. A full size prototype of a low mass drift chamber for the inner part of the HADES spectrometer has been developed. The prototype has been tested under laboratory conditions with ^{55}Fe and ^{90}Sr sources at JINR and GSI as well as in the beam of the SIS. A spatial resolution about 70 microns σ was obtained with $\text{He}-i\text{C}_4\text{H}_{10}$ (60:40) gas mixture.

The investigation has been performed at the Laboratory of High Energies, JINR.

1. ВВЕДЕНИЕ

Как показывают современные КХД-расчеты, сигналом образования плотной и горячей ядерной материи (так же, как и возможного образования кварк-глюонной фазы) может служить изменение свойств адронов (массы, константы распада, формфакторы и др.) внутри такой среды. Наиболее полную информацию о таких изменениях дает исследование электромагнитных мод распада адронов, так как лептоны и фотоны слабо взаимодействуют с ядерной материей. Изучение таких распадов позволяет получить информацию о свойствах распадающихся адронов на всех стадиях образования плотной и горячей ядерной материи вплоть до перехода в нормальное состояние. Исследование изменений свойств диэлектронных мод распада легких векторных мезонов (ρ , ω , ϕ), образовавшихся при столкновении различных ядер при энергиях 1 – 2 ГэВ/нуклон, может свидетельствовать об образовании горячей и плотной ядерной материи и позволит изучать свойства такой материи. По различным теоретическим оценкам, достигаемое при этих энергиях увеличение плотности ядерной материи может приводить к уменьшению масс этих резонансов на $\sim 15 - 30\%$ и (или) увеличению ширины распадов.

Регистрация и выделение диэлектронных мод распада адронов, доля которых составляет $10^{-5} - 10^{-7}$ от адронных мод распада, на уровне $\sim 10^{-3}$ от адронного фона является сложной методической задачей. Такими уникальными возможностями будет обладать установка HADES (High Acceptance Di-Electron Spectrometer), создаваемая коллаборацией ученых из 19 институтов 9 европейских стран.

Рождение диэлектронных пар в ядро-ядерных столкновениях исследовалось в двух экспериментах: на установке DLS в BNL [1] и на установке CERES в ЦЕРНе [2,3]. Полученные на этих установках экспериментальные данные о рождении e^+e^- -пар позволили обнаружить существенные отличия спектров эффективных масс диэлектронных пар от теоретически рассчитанных.

Установка HADES будет иметь значительно лучшие (чем в [1,2]) аксептанс и разрешение по эффективным массам дилептонных пар. Разрешение установки HADES позволит разделить ω - и ρ -мезоны, имеющие близкие массы. Скорость набора статистики будет примерно в 25 раз выше, чем на DLS. Кроме того, использование ускорителя SIS (GSI, Дармштадт) позволит исследовать рождение диэлектронных пар в столкновениях адронов от π^+p до $^{238}\text{U}^{238}\text{U}$ при кинетических энергиях пучка до 2 ГэВ/нуклон.

2. СПЕКТРОМЕТР HADES

Схематично спектрометр HADES представлен на рис.1 [4]. Основными элементами установки являются газовый черенковский детектор, многослойные дрейфовые камеры,

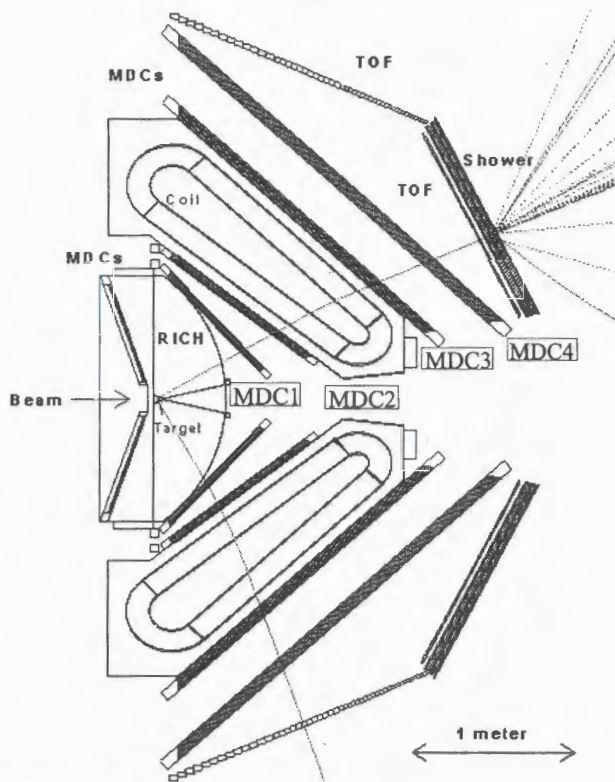


Рис. 1. Схематичный вид спектрометра HADES: RICH — черенковский счетчик, MDCs — многопроволочные дрейфовые камеры, TOF — времяпролетные сцинтилляционные детекторы, Shower — детектор электромагнитных ливней

сверхпроводящие магниты, времяпролетные детекторы и детекторы электромагнитных ливней. Спектрометр покрывает полярный угол от 18 до 85 градусов и полный азимутальный угол и имеет геометрический аксептанс около 40% [5]. Черенковский детектор (RICH) располагается вокруг мишени и используется для идентификации электронов. Он состоит из газового радиатора, сферического зеркала и быстрого газового фотонного детектора с фотокатодом из CsI [6]. Информация с черенковского детектора используется для выработки триггера второго уровня для отбора событий, в которых родились e^+e^- -пары. Набор из четырех плоскостей многопроволочных дрейфовых камер (MDCs) совместно с шестью сверхпроводящими магнитами (Coil) образуют магнитный спектрометр для измерения импульсов заряженных частиц. Прецизионные многослойные дрейфовые камеры с малым количеством вещества, имеющие пространственное разрешение ~ 70 мкм, и сверхпроводящие магниты позволяют определить импульсы частиц с точностью $\sim 1\%$, что обеспечит разрешение по эффективным массам $\sim 1\%$ в районе масс ρ - и ω -мезонов. За магнитным спектрометром расположены времяпролетные сцинтилляционные детекторы (TOF) и набор детекторов электромагнитных ливней (Shower), расположенных в полярном угле от 18 до 45 градусов. Они дают информацию о множественности рождения заряженных частиц и электрон-позитронных пар при взаимодействии пучка с мишенью, которая используется для выработки триггеров первого и второго уровней.

Времяпролетные детекторы с временным разрешением ≈ 150 пс позволят надежно идентифицировать регистрируемые частицы. Ливневый детектор [7] состоит из трех камер в каждом секторе, разделенных свинцовыми конверторами толщиной 1,1 см. Все камеры идентичны и работают в самогасящемся стримерном режиме на смеси $\text{Ag}-\text{iC}_4\text{H}_{10}-\text{C}_7\text{H}_{16}$ (66-33-1).

Эксперименты будут проводиться на ускорителе SIS (Дармштадт) с пучками различных ядер (до ^{238}U включительно). Расчетная скорость набора статистики за один день облучения установки (для $\text{Au} + \text{Au}$ взаимодействий при $T = 1$ ГэВ): 5000 событий с рождением ρ -мезона, 1000 — ω и 100 — ϕ . Создаваемая установка по своим параметрам не имеет аналогов в мире.

3. ТРЕКОВАЯ СИСТЕМА СПЕКТРОМЕТРА

Для восстановления треков заряженных частиц используется магнитный спектрометр, состоящий из тороидального магнита и 4 систем дрейфовых камер в форме усеченной пирамиды. Магнит, смонтированный из 6 сверхпроводящих катушек толщиной 80 мм каждая, создает поле 0,7 Тл с плотностью тока 120 А/мм² на каждую катушку. Требуемое разрешение по импульсу может быть получено путем оптимизации как собственного пространственного разрешения дрейфовых камер трековой системы, так и ограничением многократного рассеяния, влияющего на траекторию электронов.

Полная чувствительная площадь всех четырех систем многопроволочных дрейфовых камер, предназначенная для регистрации электронов, а также и других заряженных частиц, составляет 24 м². Две системы размещены перед магнитом, а две другие — за ним. Каждая система состоит из 6 модулей. При этом боковые стороны прижимных рам каждого модуля располагаются в области тени от катушек магнита. Каждый модуль имеет форму трапеции. Для того чтобы реконструировать и подобрать пары двух частей треков каждой частицы с минимальной неоднозначностью, каждый модуль включает в себя 6 слоев с чувствительными и потенциальными проволоками, ориентированными под углами -40° , $+20^\circ$, 0° , 0° , -20° , $+40^\circ$ градусов по отношению к основанию трапеции (рис.2). Все катодные плоскости намотаны под углом 90° и разделяют анодные плоскости. Размер ячейки увеличивается от 5×5 мм² в камерах первой системы, ближайшей к мишени, до 14×10 мм² в камерах четвертой системы. Таким образом обеспечивается хорошая и однородная гранулярность вдоль всего спектрометра.

Ширина корпуса катушек сверхпроводящего магнита составляет всего 80 мм и определяет ширину боковых сторон дрейфовых камер. Это означает, что ширина боковых сторон камер для первой и второй систем равна 30 и 39 мм, соответственно. Рамы из стеклотекстолита, на которые намотаны проволоки, с такими размерами боковых сторон не способны компенсировать силы натяжения проволок. Прогиб боковых сторон приводит к уменьшению чувствительной площади камер. Использованы различные методы для компенсации этого прогиба. В модулях первой системы применены дополнительные рамки из нержавеющей стали, которые несут всю нагрузку от натяжения проволок. Для модулей второй системы боковые стороны рамок изготавливались криволинейными. Величина изгиба каждой стороны была рассчитана и проверена экспериментально. После переноса проволок на стеклотекстолитовую рамку величина прогиба боковой стороны не превышала 0,5 мм.

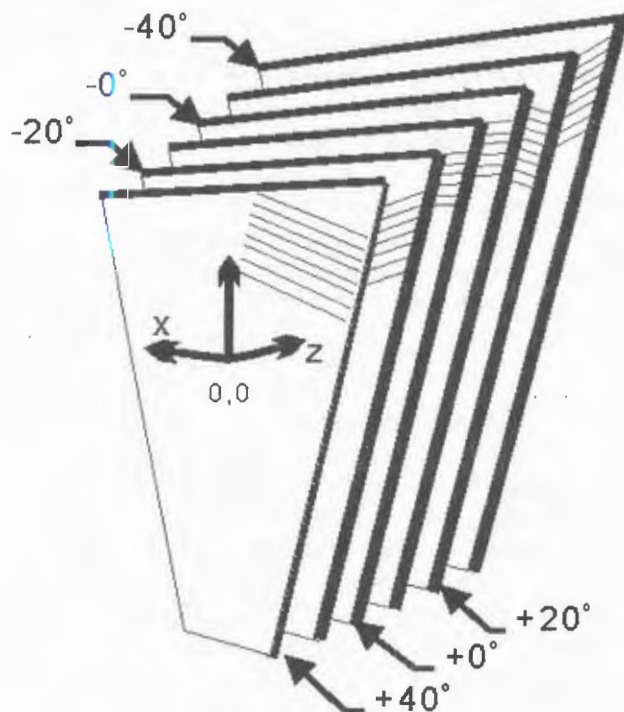


Рис. 2. Схематичный вид дрейфовой камеры, показывающий 6 чувствительных слоев. Вертикально намотанные катодные плоскости, разделяющие анодные плоскости, не показаны

Для того чтобы корректно восстановить импульс электрона и позитрона, необходимо принять во внимание многократное рассеяние вдоль всего спектрометра. Многократное рассеяние доминирует при импульсах ниже 0,4 ГэВ/с и может быть сведено к минимуму при использовании материалов с малой массой внутри камеры и наполнении гелием области магнитного поля между второй и третьей системами камер. Для анодных проволок используется 20 мкм золоченная проволока, для катодных и потенциальных — алюминиевая проволока диаметром 80 мкм (частично для модулей, находящихся за магнитом, будет использована 100 мкм алюминиевая проволока). Недостатком алюминиевой проволоки является ее текучесть. В течение первых нескольких недель натяжение алюминиевой проволоки падает на 15% и затем стабилизируется. Для входного окна применяется алюминизированный майлар толщиной 12 мкм. В качестве рабочего газа используется смесь гелий — изобутан. Вклад многократного рассеяния от различных материалов, присутствующих в активном объеме дрейфовой камеры, представлен на рис.3.

4. ПОЛНОМАСШТАБНЫЙ ПРОТОТИП ДРЕЙФОВОЙ КАМЕРЫ

С целью разработки технологии изготовления дрейфовых камер с малым количеством вещества в ЛВЭ ОИЯИ был создан полномасштабный прототип многопроволочной дрейфовой камеры (MDC-2) (рис.4) [8]. По своим размерам он полностью соответствует

[60:40]



Рис. 3. Вклад многократного рассеяния от различных материалов, присутствующих в активном объеме дрейфовых камер

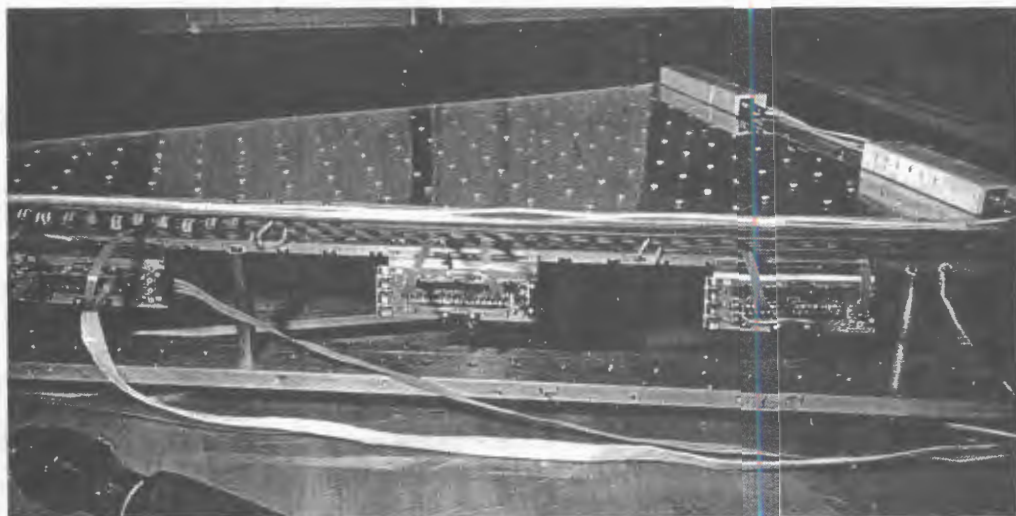


Рис. 4. Прототип дрейфовой камеры

размерам модуля второй системы и имеет трапецидальную форму. Шесть чувствительных слоев, на которых анодные и потенциальные проволоки располагаются под углами $+40^\circ$, -20° , 0° , 0° , $+20^\circ$, -40° по отношению к основанию трапеции, разделены катодными плоскостями, намотанными под углом 90° . С целью повышения жесткости анодные и катодные плоскости склеены попарно и образуют "суперслой". Для сигнальных проволок применялась золоченная вольфрамовая проволока диаметром 30 мкм. Потенциальные и

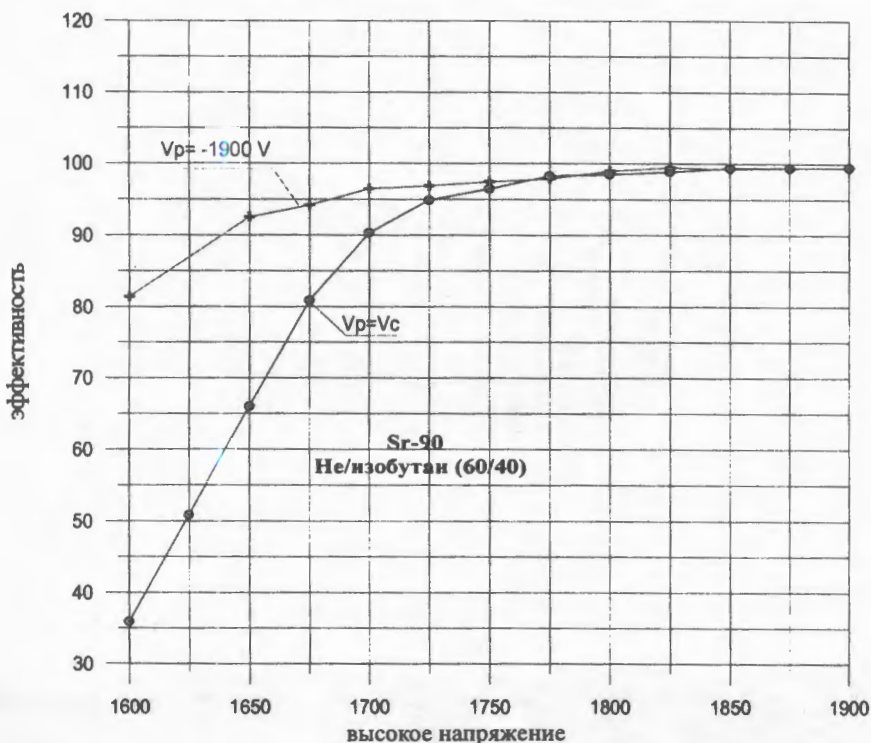


Рис. 5. Плато счетной характеристики при различных напряжениях на потенциальных проволоках

катодные проволоки намотаны 100 мкм алюминиевой проволокой. Размер дрейфовой ячейки $5 \times 6 \text{ мм}^2$.

Испытания прототипа были проведены как в лабораторных условиях с источниками ^{55}Fe и ^{90}Sr , так и на пучке 2,1 ГэВ/с протонов на синхротроне тяжелых ионов (SIS) в GSI. На рис.5 показано плато эффективности, измеренное с помощью коллимированного источника ^{90}Sr и сцинтиляционных счетчиков на газовой смеси гелий — изобутан. Повышенная концентрация изобутана увеличивает плато эффективности и улучшает временное разрешение детектора. Это обусловлено более высокой первичной ионизацией, которая возрастает с увеличением концентрации гасителя. Кроме того, при низкой концентрации гасителя плато эффективности сужается из-за большого количества ультрафиолетовых фотонов, образующихся в лавине при высоком коэффициенте усиления. Газовая смесь с соотношением 60% гелия и 40% изобутана была выбрана нами как наиболее оптимальная.

Разброс коэффициента усиления по всей площади детектора и для различных чувствительных слоев при облучении источником ^{55}Fe не превышал 8%.

С целью дальнейших испытаний прототип облучался 2,1 ГэВ/с протонным пучком на синхротроне тяжелых ионов в GSI. Два двухкоординатных микростриповых детектора [9], 300 мкм толщиной каждый, с 50 мкм шагом стрипов и чувствительной площадью $3,2 \times 3,2 \text{ см}^2$ были использованы в качестве внешнего трекера. Пространственное разрешение $< 80 \text{ мкм}$ (σ) было получено на более чем 80% дрейфовых ячеек (рис.6). Практически линейная корреляция достигается между временем дрейфа электронов и положением

трека. Скорость дрейфа составляет 4,2 см/мкс для газовой смеси $\text{He}-i\text{C}_4\text{H}_{10}$ (60% – 40%) при напряжениях на потенциальных и катодных проволоках выше 1,9 кВ. Измерения регулярности шага анодных проволок для различных сигнальных слоев показали, что смещение положения анодных проволок от расчетной величины не превышает 20 мкм.

Ожидаемый максимальный собираемый заряд для дрейфовых камер составит примерно 10 мКл на сантиметр чувствительной проволоки в год. Предполагается, что эксперимент будет продолжаться в течение 10 лет. Данная величина заряда хотя и является достаточно умеренной, требует осторожности в выборе материалов для конструкции рамок камер и эпоксидной смолы, чтобы предотвратить отравление рабочей смеси и уменьшить старение камер. Тесты по старению были выполнены на одном из макетов [8]. Измерения усиления делались путем периодической записи амплитуды сигнала в одной ячейке, которая постоянно облучалась источником ^{55}Fe в течение 40 дней. Очень незначительное старение камеры проявляется после того, как доза, полученная детектором, эквивалентна дозе, набранной в течение двух лет работы установки HADES.

5. ЭЛЕКТРОНИКА СЧИТЫВАНИЯ

Параллельно с созданием MDC-2 в ЛВЭ была проведена разработка аналоговой электроники считывания информации. Основные результаты получены при разработке первой 8-канальной версии усилителя-дискриминатора (УД) AD8 - H2 [10], изготовленной в ОП

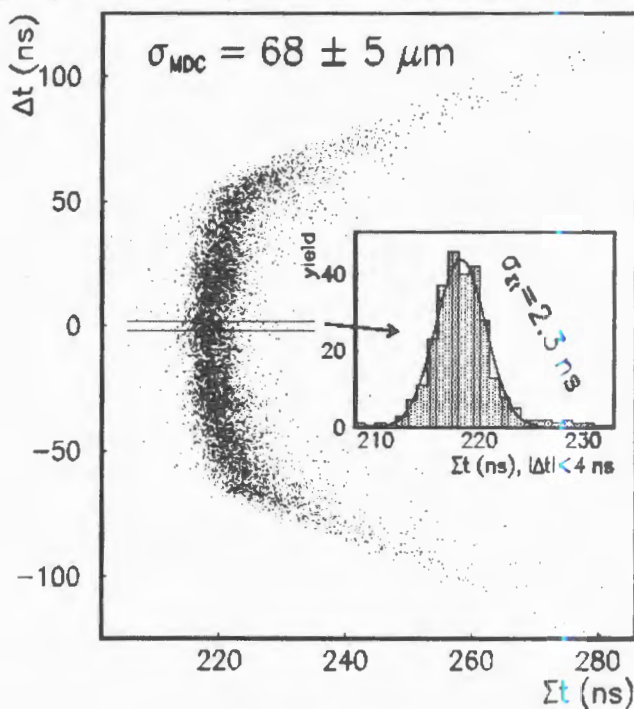


Рис. 6. Зависимость разницы — времен дрейфа, измеренных в смежных ячейках для 2 ГэВ/с протонов, от их суммы. Внутри показана проекция при $\Delta t = 0$

ОИЯИ. С помощью этой электроники проведены первые испытания полномасштабного прототипа MDC-2 на пучке протонов в GSI [8]. Была показана возможность получения с применением этой электроники высокого (~ 70 мкм) пространственного разрешения и хорошей эффективности регистрации при использовании газовой смеси на основе гелия. Окончательный 16-канальный вариант электроники с более высокой степенью интеграции разработан и испытан совместно с GSI и отдан в массовое производство (~ 30 тыс. кан.). Часть модифицированных плат с возможностью мониторинга аналоговых сигналов и цифрового ИЛИ изготовлена в ОИЯИ. Платы УД (дочерние) размещаются на материнских, на которых расположена кодирующая электроника: 8-канальные микросхемы быстродействующих времяцифровых преобразователей TDC, а также аппаратура считывания цифровых данных, контроля и задания напряжения порогов для УД.

К характеристикам УД был предъявлен ряд жестких требований: малая потребляемая мощность (~ 30 мВт/кан.), малые шумы и хорошее быстродействие для обеспечения высокой точности (лучше 100 мкм), разрешение пары импульсов 50 — 100 нс. Проблема осложнялась требованием обеспечить высокую плотность размещения каналов ($\sim 2,5$ см²/кан.), несмотря на очевидность того, что канал должен быть достаточно сложным, т.к. требуемое разрешение пары импульсов может быть получено только с применением соответствующих шейперов и схем отсечки "ионного хвоста". Также понятно, что здесь предпочтительно использование биполярной технологии, чтобы обеспечить низкие шумы при малых временах формирования (передний фронт импульса на входе дискриминатора не должен превышать 10 нс). Но использование биполярной технологии в быстрых схемах сразу ставит вопрос о потребляемой мощности.

В качестве базового элемента нами была предложена и использована специализированная микросхема ASD8, разработанная в Пенсильванском университете [11]. Микросхема состоит из 8 одинаковых каналов, каждый из которых содержит быстрый малошумящий усилитель, схемы формирования коротких импульсов и дискриминатор. Эта микросхема отличается малой потребляемой мощностью и дифференциальной структурой всех звеньев канала. Последнее обстоятельство эффективно использовано для уменьшения наводок на вход со стороны выхода и со входных неэкранированных плоских каптоновых кабелей. Выходные токовые импульсы с ASD8 преобразуются в дифференциальный сигнал GTL с помощью сопротивлений нагрузки и опорного напряжения. Все 16 каналов используют одно и то же напряжение порога, поступающее с материнской платы. На входе и выходе УД используются современные многоконтактные миниатюрные разъемы для поверхностного монтажа.

Основные параметры УД следующие: размер многослойной платы 90×40 мм², число каналов — 16, потребляемая мощность — 30 мВт/кан., передний фронт сигнала с усилителя — 8 нс, собственные шумы — 1 фКл, разрешение пары импульсов с детектора — лучше 100 нс, напряжение питания — ± 3 В, опорное напряжение — + 1 В.

6. ПРОГРАММЫ АНАЛИЗА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Программы анализа данных для установки HADES создаются на языке программирования C++ на основе пакета ROOT (<http://root.cern.ch>). Обработку данных предполагается производить на ЭВМ типа IBM PC в операционной системе LINUX. Разработку программ ведут несколько групп физиков. Дубненская группа разрабатывает программы

поиска треков и восстановления параметров частиц и вершин событий, используя информацию с MDC. В настоящее время написана программа поиска кандидатов в треки для MDC, расположенных до магнита (внутренняя часть спектрометра).

Поскольку основная цель создания установки HADES — изучение центральных столкновений тяжелых ядер, установка будет работать в условиях относительно больших загрузок частицами (20—30 частиц на одну MDC). Это означает, что в каждой MDC из порядка $10^6 - 10^7$ комбинаций сработавших проволочек надо найти 20—30 истинных комбинаций, соответствующих реальным трекам. Для решения этой достаточно сложной задачи нами предложен и реализован быстрый алгоритм поиска кандидатов в треки в MDC центральной части спектрометра, позволяющий также определить, в первом приближении, геометрические параметры трека. При поиске кандидатов в треки время дрейфа не учитывается, что позволяет сделать предварительно большинство расчетов для всех событий. Алгоритм опробован на моделированных и пропущенных через программу GEANT событиях центральных $Au + Au$ -столкновений.

Алгоритм основан на двух свойствах установки. Это малые размеры мишени (около 2 см для мишени из золота) относительно расстояния до MDC (> 40 см) и приблизительно прямолинейные формы треков на участке от мишени до магнита. Основная идея алгоритма заключается в проецировании чувствительных объемов всех сработавших проволочек из MDC1 и MDC2 (12 плоскостей чувствительных проволочек в каждом секторе) на одну плоскость относительно мишени и последующем поиске на этой плоскости пиков, соответствующих пересечению сработавших проволочек для каждого трека.

Оценка этого алгоритма на моделированных событиях показала, что при загрузке 20—30 частиц на один модуль MDC программа позволяет найти все треки из мишени с хорошим качеством отбора сработавших проволочек (более 80% отобранных проволочек в пике принадлежат одному треку). Примесь ложных треков составляет не более нескольких процентов. Точность нахождения положения трека на плоскости проектирования примерно в пять раз лучше, чем размер чувствительной области сигнальной проволочки на плоскости проектирования ($\sigma = 1,1$ мм при расстоянии между чувствительными проволочками 6 мм).

Для оценки дальнейших шагов по разработке программ восстановления треков в MDC мы попробовали профитировать треки прямой линией (с учетом времени дрейфа), закладывая в фит как начальное приближение данные, полученные вышеописанным методом. В результате мы получили точность восстановления треков около 100—140 мкм, что соответствует расчетным значениям разрешения MDC спектрометра HADES.

7. ЗАКЛЮЧЕНИЕ

Разработана конструкция и создана многослойная дрейфовая камера для внутренней части трековой системы спектрометра HADES. Использование газовой смеси гелий—изобутан и алюминиевой проволоки позволяет уменьшить радиационную длину детектора, что особенно важно для восстановления треков частиц с малым импульсом. Увеличение концентрации изобутана несколько увеличивает многократное рассеяние, но при этом достигается значительно более устойчивый режим работы детектора.

Испытания детектора в лабораторных условиях с источниками ^{55}Fe и ^{90}Sr в ОИЯИ (Дубна) и на пучках 2,1 ГэВ/с протонов и ^{238}U SIS в GSI (Дармштадт) показали ста-

бильную работу детектора. Пространственное разрешение ~ 70 мкм получено на газовой смеси $\text{He}-i\text{C}_4\text{H}_{10}$.

В настоящее время изготовлены все модули дрейфовых камер внутренней части спектрометра и электроника считывания информации и начаты их испытания в составе спектрометра.

Авторы выражают благодарность академику А.М.Балдину за внимание и интерес к работе. Работа, представленная в этой статье, поддерживалась INTAS, Brussels (контракты 94-1233, 96-0468), а также Минатомом (Россия) и BMBF (Германия) в рамках программы DLR/WTZ 662-97.

Авторы благодарят также Ю.Г.Федулова, Н.П.Волкова, А.Е.Московского, Н.Н.Тиханчева и В.А.Белякова за участие в создании детекторов и электроники.

Литература

1. Porter R.J. et al. — *Phys. Rev. Lett.*, 1997, v.79, p.1229.
2. Ullrich Th. et al. — *Nucl.Phys.*, 1996, v.A610, p.317.
3. Drees A. — *Nucl. Phys.*, 1996, v.A610, p.536.
4. GSI Scientific Report 1997, GSI 98-1, Darmstadt, 1998, p.196.
5. Schicker R. et al. — *Nucl. Instr. and Meth.*, 1996, v.A380, p.586.
6. Gernhauser R. Et al. — *Nucl. Instr. and Meth.*, 1996, v.A371, p.300.
7. Salabura P. Et al. — *Acta Phys.Pol.*, 1996, v.B27(1&2), p.421.
8. Garabatos C. et al. — *Nucl. Instr. and Meth.*, 1998, v.A412, p.38.
9. Colledani C. et al. — *Nucl. Instr. and Meth.*, 1996, v.A372, p.379.
10. Chernenko S.P. et al. — *JINR, E13-97-60*, Dubna, 1997.
11. Newcomer F.M. et al. — *IEEE Trans. Nucl. Sci.*, 1993, v.NS - 40, No.4, part 1, p.630.

УДК 539.173.4; 536.51

ИЗМЕРЕНИЕ ТЕМПЕРАТУРЫ УРАНОВОГО ОБРАЗЦА, ОБЛУЧАЕМОГО ВТОРИЧНЫМИ НЕЙТРОНАМИ

**В.И.Батин, Д.В.Батин¹, В.В.Борисов, Н.М.Владимирова,
В.Г.Георгиев², В.М.Дробин, В.С.Королев, М.И.Кривоносов,
С.В.Семашко, Л.С.Спасов², С.Г.Стеценко**

Проведено измерение температуры уранового образца, облучаемого вторичными частицами, образующимися в свинцовой мишени при взаимодействии с пучком протонов с энергией 1,5 ГэВ и средней интенсивностью $3,9 \cdot 10^9$ прот./с в течение 260 мин. Наблюдалось коррелированное с интенсивностью пучка изменение показаний термодатчиков на основе кварцевого резонатора. Оценка средней мощности, выделившейся в образце, сделанная на основе теплофизической модели, составляет 3,1 мВт и в пределах погрешности совпадает с величиной средней мощности $(3,4 \pm 0,5)$ мВт, полученной на основе данных детектора осколков деления. Чувствительность измерительного канала — 1 мК.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Temperature Measurement of the Uranium Sample Irradiated with Secondary Neutrons

V.I.Batin et al.

The measurement of temperature of the uranium sample irradiated with secondary particles, formed in a lead target, is carried out. The target was irradiated with a proton beam with the energy of 1.5 GeV and intensity $3.9 \cdot 10^9$ p./s during 260 minutes. The thermometers on the basis of quartz resonators have fixed changes of the uranium sample temperature, which correlated on time with beam intensity. The estimation of the average energy allocated in the sample, by using these data and thermophysical model, is made; it is equivalent to 3.1 mW. This value coincides with the value of the average energy of (3.4 ± 0.5) mW, obtained by using the solid state track detector. The sensitivity of the measuring channel was 1 mK.

The investigation has been performed at the Laboratory of High Energies, JINR.

¹Московский инженерно-физический институт

²ИФТТ БАН, София

1. ВВЕДЕНИЕ

При исследовании физических процессов в урановых и ториевых мишенях, облучаемых высокоэнергетическими частицами и ядрами, одним из параметров, интересующих исследователей, является количество теплоты, выделившееся в делящемся материале [1,2,10]. Для определения этого параметра, как правило, используются различные варианты калориметрических методов [3], суть которых сводится к регистрации динамического изменения температуры делящегося материала. Надо отметить, что проведение подобного рода исследований сопряжено с достаточно серьезными трудностями, т. к. из-за ограничений по интенсивности первичного пучка необходимо регистрировать очень малые ($\sim 10^{-2}$ К) изменения температуры, и поэтому термодатчики должны иметь высокую чувствительность.

Наиболее высокими метрологическими характеристиками обладают кварцевые термодатчики [4,5,6]. В этих приборах используется свойство некоторых типов кварцевых резонаторов изменять собственную частоту при изменении температуры. Кристалл резонатора в этом случае играет двоякую роль. Он является частью с одной стороны частото-задающей цепи генератора, а с другой — термочувствительного элемента. Конструктивно датчик представляет собой пластинку кварца, помещенную в стальной корпус, который, для увеличения тепловой проводимости между резонатором и корпусом датчика, заполнен азотом или гелием. Чувствительный элемент с помощью экранированного кабеля соединяется с электронной схемой, обеспечивающей возбуждение резонатора и съем сигнала с датчика.

Цель данной работы состоит в проверке методики прямого измерения малых тепло-выделений с помощью кварцевых термодатчиков в условиях интенсивного фона вторичных частиц (γ , n и т.д.).

Использованные в данной работе термодатчики имеют следующие параметры:

Резонансная частота	— 26,6 МГц (300 К)
Чувствительность	— ~ 1000 Гц /К°
Порог чувствительности	— 0,0001 К
Нелинейность выходной хар-ки	— $\sim 0,07$ К°
Масса датчика	— $(0,950 \pm 0,01)$ г
Габаритные размеры	— $(11 \times 13,5 \times 3,5)$ мм

*В диапазоне 233–425 К

2. ОПИСАНИЕ УСТАНОВКИ

Для проверки методики калориметрических измерений с термодатчиком на основе кварцевого резонатора был разработан детектор (рис. 1) и проведено измерение температуры облучаемого уранового образца детектора в сеансе синхрофазотрона в июне 1998 г. Алюминиевый цилиндр с металлическим ураном естественной смеси был помещен в двухслойный термостат из пенопласта. На торце цилиндра располагался термодатчик Т1. Между внешним и внутренним кожухом термостата находился второй, опорный, термодатчик Т2.

Детектор располагался в центре над свинцовой мишенью, помещенной в пенопластовый термостат. Вся сборка была уложена в деревянный короб $\sim 1 \times 1 \times 1$ м (рис. 2,3), обшитый изнутри листовым кадмием и заполненный гранулированным полиэтиленом.

Во время сеанса мишень облучалась пучком протонов синхрофазотрона ЛВЭ с энергией 1,5 ГэВ/с и интенсивностью $3,9 \cdot 10^9$. Облучение длилось 4 часа 20 мин, полный интеграл $4,8 \cdot 10^{13}$ протонов. Показания датчиков регистрировались один раз в секунду. Чувствительность измерительного канала составляет $1 \cdot 10^{-3}$ К. Измерения начались за 1,5 ч до начала облучения и закончились через 25 мин после его окончания.

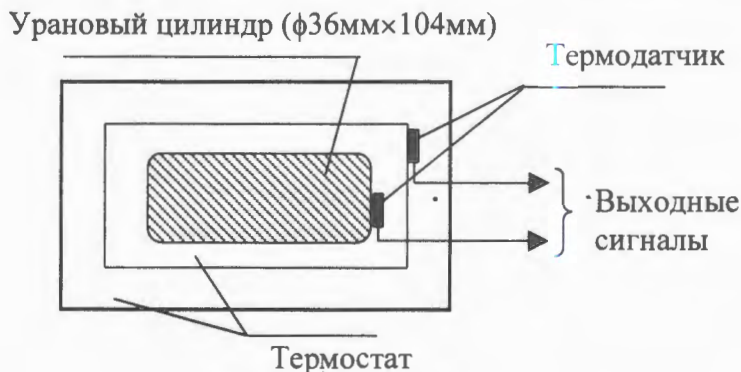


Рис. 1. Эскиз детектора

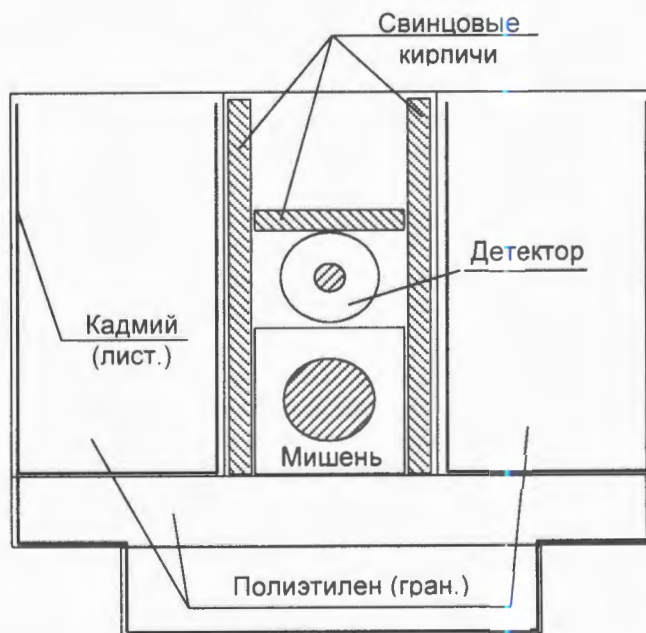


Рис. 2. Эскиз установки

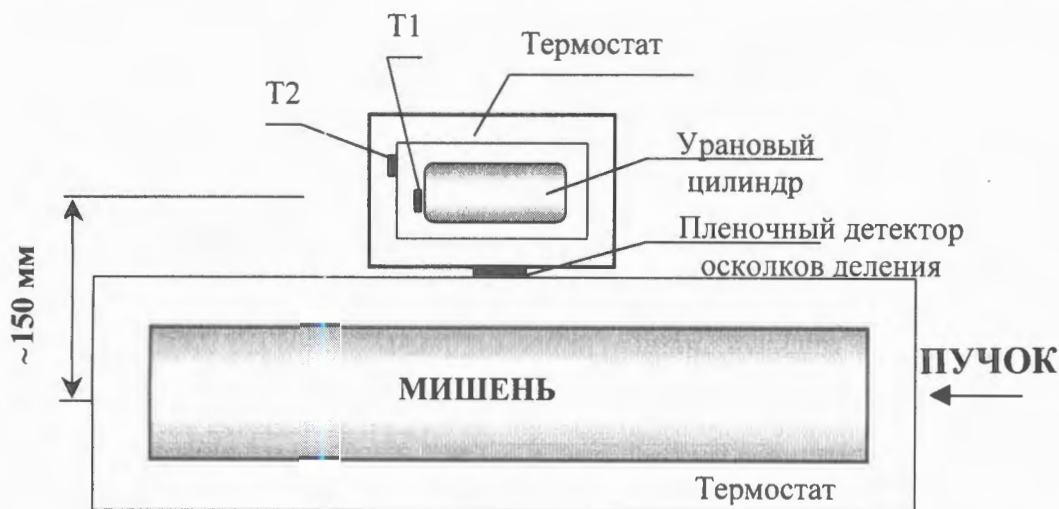


Рис. 3. Расположение детектора относительно мишени (вид сбоку, детали установки не показаны)

3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

Графики изменения температуры, зарегистрированной датчиками T1 и T2 совместно с данными об интенсивности пучка, приведены на рис. 4. Нормированный график интенсивности иллюстрирует характер поведения пучка во время сеанса облучения мишени. Графики T1 и T2 показывают изменение температуры относительно начальной точки. При этом график T1 показывает изменение температуры поверхности уранового образца детектора, а график T2 — изменение температуры внутри кожуха термостата. Разность температур T1 и T2 в момент начала измерений составляла $\sim 0,5^\circ \text{C}$ (520 мК). Детектор до размещения в установке в течение нескольких дней находился в более теплом помещении и был установлен примерно за три часа до начала облучения. Поэтому к началу измерений детектор не находился в тепловом равновесии с установкой, в которой температура была ниже. Уменьшение температуры, зарегистрированное до начала облучения обоими датчиками, вызвано процессом теплообмена детектора и установки. Кроме этого, измерения проводились на фоне общего понижения температуры окружающей среды (время измерений $\sim 15 \text{ ч} — 21 \text{ ч } 30 \text{ мин}$).

Графики температур T1 и T2 показывают, что с началом облучения падение температуры в детекторе замедлилось. При этом изменение температуры поверхности уранового образца коррелирует с изменением интенсивности пучка (рис. 4). Наблюдаемое замедление падения температуры может быть следствием:

- изменения характеристик датчика, вызванное его облучением;
- тепловыделения в уране вследствие его деления.

Оба датчика находятся в потоке вторичных частиц: нейтронов, заряженных частиц и γ -излучения, которые влияют на параметры датчиков. Однако, поскольку датчики T1 и T2 находятся в одинаковых условиях (физически расположены в непосредственной близости один от другого), изменение их параметров, вызванное облучением, должно быть одного порядка.

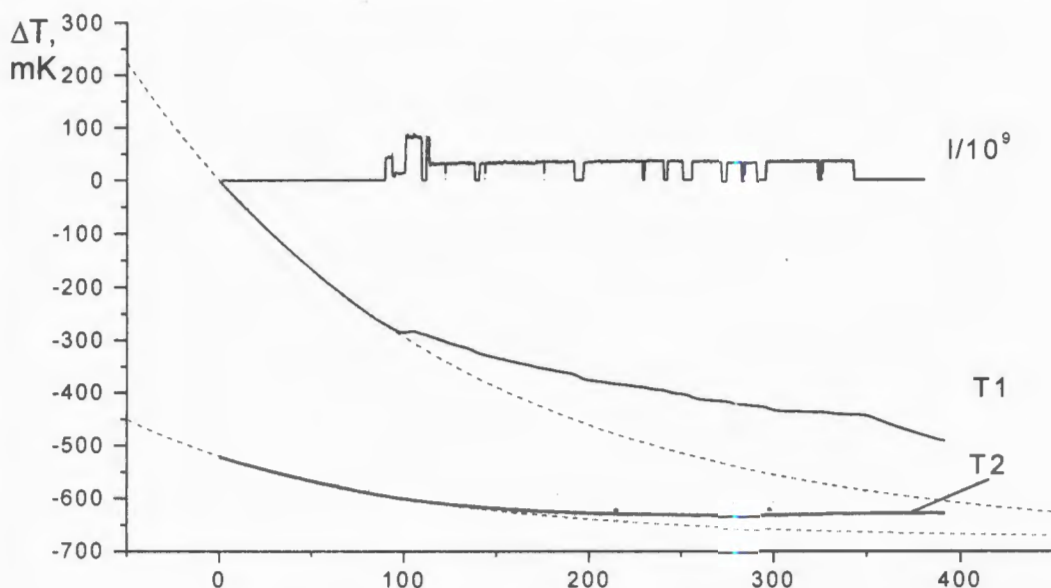


Рис. 4. Графики изменения температур T1 и T2

Влияние нейтронных потоков на кварцевые резонаторы изучено достаточно хорошо [8], и, в зависимости от типа резонатора, это влияние может приводить как к увеличению, так и к уменьшению собственной резонансной частоты прибора, вызывать обратимые и необратимые изменения в структуре кристалла. Все доступные данные, приводимые в литературе, говорят о том, что существенные изменения в структуре кварца происходят при нейтронных потоках $> 10^{10} \text{ л/}(\text{см}^2 \cdot \text{с})$, а в нашем случае предварительная оценка показывает, что максимальный поток нейтронов должен составлять не более $10^6 - 10^7 \text{ л/}(\text{см}^2 \cdot \text{с})$. По-видимому, величина воздействия нейтронного потока на датчики в данном случае лежит ниже порога чувствительности измерительного тракта.

Оценить с достаточной точностью влияние заряженных частиц и γ -излучения на датчики в данный момент не представляется возможным, и поэтому полностью исключить наличие такого влияния нельзя. Однако отсутствие явной корреляции показаний датчика T2 с интенсивностью позволяет сделать предположение, что такое влияние мало и не вызывает заметных изменений.

Сравнивая между собой показания датчиков, мы видим, что у датчика T1 наблюдается отклик на скачки интенсивности, у датчика T2 такого отклика не наблюдается. Если бы у датчиков изменились их характеристики, то такое изменение было бы одинаковым для обоих датчиков. Таким образом, можно считать, что регистрируемые изменения показаний датчиков имеют тепловую природу.

4. ОЦЕНКА МОЩНОСТИ, ВЫДЕЛЯЕМОЙ В УРАНОВОМ БЛОКЕ, ПО РЕЗУЛЬТАТАМ ИЗМЕРЕНИЙ

Хотя измерения проводились только в одной точке поверхности уранового образца и в одной точке внутри термостата, анализ показывает, что для получения оценки средней

мощности температуру поверхности уранового образца можно считать одинаковой для всех точек, и плотность потока тепла с поверхности также можно считать одинаковой по всей поверхности.

От времени сборки системы до начала измерения прошло около 3,5 ч, за это время нестационарный процесс теплопроводности должен был перейти в регулярный режим, при котором температура $T(x, y, z, t)$ любой точки внутри тела или системы тел изменяется по закону

$$T(x, y, z, t) = T_c + A \cdot U(x, y, z) \cdot e^{-mt},$$

где T_c — температура на границе системы.

Температура $\Delta T = T - T_0$, отсчитываемая от некоторой начальной температуры T_0 , тогда будет иметь следующий вид:

$$\Delta T = T_c - T_0 + A \cdot U(x, y, z) \cdot e^{-mt},$$

поскольку в начальный момент $\Delta T = 0$, следовательно, и окончательно:

$$A \cdot U(x, y, z) = T_c - T_0,$$

где m — темп регулярного режима.

Приведенное выражение находится в хорошем согласии с экспериментальными данными для временного интервала, предшествующего облучению, $m_2 = (8,25 \pm 0,09) \cdot 10^{-3}$, $T_c - T_{02} = 141,5 \pm 1,3$; $m_1 = (5,92 \pm 0,02) \cdot 10^{-3}$, $T_c - T_{01} = 656 \pm 2$.

В стадии регулярного режима распределение температуры в урановом образце можно считать равномерным, и поэтому можно положить температуру поверхности равной средней по объему температуре урана. Далее везде считается $T_s = \langle T \rangle_v = T$. Уравнение теплового баланса в этом случае:

$$C \cdot \frac{\partial T}{\partial t} + q_n \cdot S = P,$$

где C — теплоемкость образца; q_n — плотность потока тепла с поверхности блока; S — площадь поверхности; P — мощность, выделяемая в блоке. В отсутствие источников тепла

$$q_n = -\frac{C}{S} \cdot \frac{\partial T}{\partial t}.$$

Используя регулярный режим, получаем для q_n :

$$q_n = -m \cdot \frac{C}{S} \cdot (\Delta T - T_c + T_0).$$

И далее

$$\frac{\partial \Delta T}{\partial t} + m \cdot (\Delta T - T_c + T_0) = \frac{P}{C}.$$

Решая это уравнение, находим для ΔT :

$$\Delta T = (T_c - T_0) \cdot (1 - e^{-mt}) + \frac{P}{mC} \cdot (1 - e^{-m(t-t_0)});$$

t_0 — время начала выделения мощности.

Применяя полученную зависимость к экспериментальным данным, мы находим среднюю мощность, выделяемую в уране, усредненной по всему интервалу времени облучения $P = 3,1$ мВт.

Показания датчика Т2, а также «тонкая структура» в графике Т1, вызванная скачками интенсивности, качественно и количественно объясняются приведенной моделью.

5. РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ ДЕТЕКТОРА ОСКОЛКОВ ДЕЛЕНИЯ

Для проверки результатов, полученных с помощью теплофизической модели, произведен расчет энергии, выделившейся в урановом образце за счет деления ядер урана. При этом для определения количества событий деления использовался детектор, состоящий из урановой фольги толщиной 50 мкм того же изотопного состава, что и уран образца, и находящейся с ней в плотном контакте лавсановой пленки толщиной 160 мкм (регистрируются с 90% эффективностью осколки деления ядер урана, вылетевшие в 2π). Определение количества событий деления ядер урана в образце проводилось в предположении, что поток нейтронов и других, вызывающих деление ядер, частиц, одинаков по энергетическому спектру и в области уранового образца, и пленочного детектора, а по интенсивности отличается на отношение квадратов расстояний от оси пучка. Использование такого детектора, при вышеуказанном допущении, позволяет определить количество событий деления без энергетического спектра нейтронов и измерения сечений деления урана нейтронами различных энергий, а также без измерения интенсивности первичного пучка протонов.

Для определения числа событий деления и расчета энерговыделения использованы следующие данные:

Вес образца		1720 г
Высота		104 мм
Диаметр		36,5 мм
Плотность урана	ρ	19,04 г/см ³
Расчетная толщина алюминиевого покрытия		1,65 мм
Вес алюминиевого покрытия		58,4 г
Вес урана в образце		1659 г
Длина блока металлического урана	L	100,7 мм
Диаметр	D	33,2 мм
Эффективно работающий слой урана	d	$4,5 \cdot 10^{-3}$ г/см ²
Плотность треков в детекторе	N_1	$1,13 \cdot 10^6$ тр./см ²
Эффективность регистрации осколков	α	0,7
Расстояние от центра пучка до детектора	R_1	85 мм
Расстояние от центра пучка до центра образца	R_2	150 мм
Величина энерговыделения в одном делении	K	$2,59 \cdot 10^{-11}$ Дж
Время облучения	Δt	$1,41 \cdot 10^4$ с

Количество событий деления ядер урана:

$$N_{\text{дел}} = \pi \cdot \left(\frac{R_2}{R_1} \right)^2 \cdot \frac{D^2}{4} \cdot \frac{N_1}{\alpha} \cdot \frac{L\rho}{d},$$

и мощность, выделившаяся в урановом образце:

$$P = \frac{N_{\text{дел}} K}{\Delta t}.$$

Величина энерговыделения при одном событии деления урана принималась равной 162 МэВ — энергии осколков деления, всеми остальными источниками энерговыделения (η , γ и т.д.) пренебрегали из-за небольшого размера уранового образца и малого времени измерения. Величина средней мощности, полученная на основе данных детектора осколков деления, составляет $(3,4 \pm 0,5) \cdot 10^{-3}$ Вт.

6. ВЫВОДЫ

Совпадение результатов, полученных двумя независимыми методами, позволяет сделать вывод о том, что предложенный способ прямого измерения малых тепловыделений в условиях интенсивного фона вторичных частиц вполне работоспособен и показывает хорошие результаты.

Большой запас по чувствительности детектора позволяет существенно (в 20–30 раз) уменьшить массу уранового образца и соответственно уменьшить габаритные размеры детектора.

Предложенная методика позволяет проводить on-line измерения тепловыделений при существенном уменьшении времени облучения, с заранее заданной требуемой точностью. Это утверждение иллюстрирует график зависимости относительной погрешности определения средней мощности от времени измерения (рис. 5).

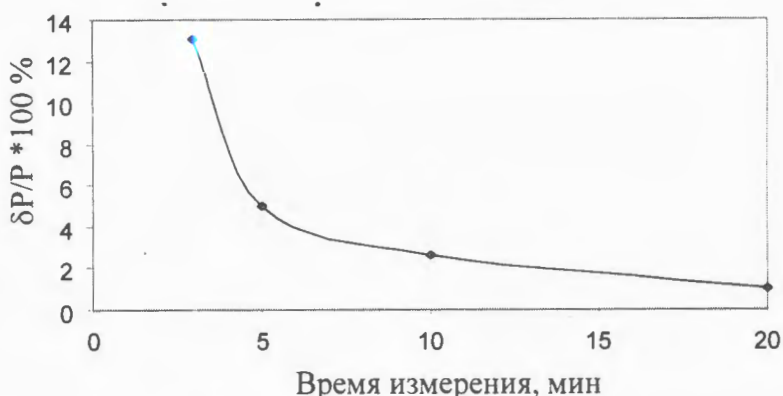


Рис. 5. Зависимость величины ошибки от длительности временного интервала

Авторский коллектив приносит глубокую благодарность руководству отдела Физики многочастичных систем ФИАН (отдел «Дубна») за поддержку данной работы, а также профессору И.А.Шелаеву и профессору С.А.Хорозову за очень полезные обсуждения и рекомендации по обработке экспериментального материала.

Литература

1. Михайлов В.Н. — Краткие сообщения ОИЯИ, 1996, №6[80]-96, с.17.
2. Rubbia C. — Physics Letters, 1995, v.B348, с.697.
3. Ашш Ж. и др. — Датчики измерительных систем. М.; Мир, 1992.
4. Малов В.В. — Пьезорезонансные датчики. М.; Энергия, 1978.
5. Спасов Л., Дробин В. и др. — Препринт ОИЯИ, 1996.
6. Кварцевый термопреобразователь ТПР-К. Техническое описание и паспорта датчиков № 393, Н028, Н032, Н002, Н024, 825.
7. Баков А. и др. — Атомная энергия, 1980, т.48 вып.1, с.39.
8. Влияние облучения на материалы и элементы электронных схем. (пер. с англ.) Под ред. В.Н.Быкова и С.П.Соловьева, М.: Атомиздат 1967.
9. Юревич В.И. и др. — Краткие сообщения ОИЯИ, 1998, №3[89]-98, с.65.
10. Проблемно-тематический план научно-исследовательских работ и международного сотрудничества ОИЯИ на 1999 год.

УДК 539.1.074.23

КРАЕВЫЕ ЭФФЕКТЫ В ПРОПОРЦИОНАЛЬНОЙ КАМЕРЕ. ВЛИЯНИЕ ДИЭЛЕКТРИЧЕСКОЙ РАМКИ

Т.П.Топурия, М.Д.Шафранов

Представлены электрические характеристики пропорциональных камер на границах с рамкой камеры. Применявшиеся до сих пор методы расчета электрических полей и потенциалов в рабочем объеме многопроволочных пропорциональных камер не позволяли оценить степень влияния диэлектрической рамки камеры на формирование электрических полей в районе крайних проволочек. Решение поставленной задачи стало возможным только после разработки авторами настоящей публикации методов решения задач о диэлектрике в электрическом поле.

Работа выполнена в Лаборатории физики частиц ОИЯИ.

Edge Effects in Multiwire Proportional Chambers. The Influence of the Dielectric Frame

T.P.Topuria, M.D.Shafranov

The electrical characteristics of proportional chambers on boundaries with a frame of the chamber are presented. The earlier methods of calculating the field and potentials in the working volume of multiwire proportional chambers did not allow one to estimate the influence of the dielectric frame of the chamber on the configuration of electrical field in the area of the edge wires. This simulation can be carried out now since the authors of the present publication have developed the methods of solving the problem of dielectric in electrical field.

The investigation has been performed at the Laboratory of Particle Physics, JINR.

ВВЕДЕНИЕ

Исследование краевых эффектов в многопроволочных пропорциональных камерах (МППК) существенно для понимания процессов в камере как координатном детекторе частиц, а также для выбора ее оптимальной конструкции. Краевые эффекты в МППК проявляются в росте числа шумовых импульсов на крайних проволочках камеры и даже в появлении пробоев.

Краевые эффекты связаны с увеличением электрического поля в районе крайних проволочек камер. Электрическое поле непосредственно вокруг проволочки пропорционально линейной плотности ее заряда. Увеличение линейной плотности заряда на крайних проволочках является следствием роста величины линейной емкости крайних

проволочек. В стандартной конструкции МППК площадь катодных плоскостей всегда больше площади, занимаемой проволочками. Такова первая причина, которая приводит к росту линейной емкости проволочек на краях МППК. Часть диэлектрической рамки камеры частично расположена между катодными плоскостями, образуя вместе с ними конденсатор с диэлектриком между обкладками. Это вторая причина, которой обязано увеличение линейной емкости проволочек на краях камеры.

Настоящая работа посвящена исследованию краевых эффектов МППК, обусловленных указанными выше причинами. Изучение этих эффектов численным итерационным методом без учета влияния рамки показало, что на краях камеры происходит деформация зоны чувствительности проволочек. Вместо прямоугольной формы, свойственной центральной области, на краях камеры зона чувствительности приобретает сложную форму [1].

Моделирование влияния диэлектрической рамки на распределения электрических полей и потенциалов стало возможным благодаря применению специально разработанного численного метода решения задач о диэлектрике в электростатическом поле [2,3].

ОБ ЭЛЕКТРОСТАТИКЕ ДИЭЛЕКТРИКОВ

Решение краевой задачи указанным методом позволяет найти при заданном распределении зарядов их величины, обеспечивающие выполнение граничных условий. В качестве граничного условия на поверхностях проводников задается потенциал. Граничные условия на поверхности диэлектрика определяются законами электростатики для диэлектриков. Граничные условия на поверхности раздела между средами в электростатических полях для однородных диэлектриков определяются из двух уравнений [4]:

$$\operatorname{rot} \mathbf{E} = 0 \quad \text{и} \quad \operatorname{div} \mathbf{D} = 0, \quad (1)$$

где вектор $\mathbf{E}$ — электрическое поле как внутри, так и вне диэлектрика. Поле является суперпозицией поля сторонних зарядов $\mathbf{E}_0$ и поля связанных зарядов $\mathbf{E}^0$: $\mathbf{E} = \mathbf{E}_0 + \mathbf{E}^0$. Во втором уравнении $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$, где $\mathbf{P}$ — вектор поляризации, или электрический дипольный момент единицы объема диэлектрика; ϵ_0 — диэлектрическая постоянная. Следствием первого уравнения из (1) является непрерывность тангенциальных составляющих электрического поля на поверхности раздела двух диэлектриков:

$$E_{t1} = E_{t2}. \quad (2)$$

Следствием второго уравнения из (1) является непрерывность нормальной составляющей вектора $\mathbf{D}$ к поверхности раздела:

$$D_{n1} = D_{n2}. \quad (3)$$

Здесь n_1 и n_2 — внешние нормали к границе раздела среда — диэлектрик. Нормаль n_1 направлена от диэлектрика в среду, а нормаль n_2 — внутрь него.

Для однородных диэлектриков $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P} = \epsilon \epsilon_0 \mathbf{E}$, где ϵ — диэлектрическая проницаемость среды. В этом случае граничные условия (2) и (3) представляются в виде

соотношений для тангенциальных и нормальных составляющих полей двух однородных диэлектриков:

$$E_{t1} = E_{t2}, \quad \varepsilon_1 E_{n1} = \varepsilon_2 E_{n2}. \quad (4)$$

Граничные условия (4) можно представить и в виде граничных условий для потенциала:

$$\phi_1 = \phi_2, \quad \varepsilon_1 \partial \phi_1 / \partial n = \varepsilon_2 \partial \phi_2 / \partial n. \quad (5)$$

В нашем случае можно полагать, что диэлектрическая проницаемость среды, окружающей диэлектрическую рамку $\varepsilon_1 = 1$. Соотношения (4 и 5) запишутся как:

$$E_{t1} = E_{t2}, \quad E_{n1} = \varepsilon E_{n2} \quad \text{и} \quad \phi_1 = \phi_2, \quad \partial \phi_1 / \partial n = \varepsilon \partial \phi_2 / \partial n. \quad (6)$$

Из соотношений (6) следует закон преломления линий поля E : $\operatorname{tg} \alpha_1 / \operatorname{tg} \alpha_2 = \varepsilon$.

В реальных случаях в качестве граничных условий можно задать только потенциалы проводников. Для решения задачи необходимо знать величины тангенциальных составляющих электрического поля или потенциалов на границах диэлектрика и нормальных составляющих электрического поля.

Нормальная составляющая электрического поля связана с плотностью поляризованных зарядов на поверхности диэлектрика соотношением

$$\sigma = \kappa \varepsilon_0 E_n,$$

где диэлектрическая восприимчивость $\kappa = 1 + \varepsilon$.

Чтобы определить величину нормальной составляющей электрического поля в некоторой точке поверхности диэлектрика, необходимо знать поверхностную плотность связанных поляризационных зарядов в этой точке. Поверхностную плотность связанных поляризационных зарядов в соответствии с граничными условиями (1.6) нельзя вычислить без знания распределения величин нормальных составляющих поля на поверхности диэлектрика.

Задания потенциалов на поверхностях проводников и величин диэлектрических проницаемостей диэлектриков не достаточно для решения задач с диэлектриками в электростатических полях. Задача является неразрешимой. Предложенный в [2,3,5] простой, чисто алгебраический, метод позволяет решать широкий класс неразрешимых ранее задач о диэлектриках в электростатических полях.

Для решения поставленной задачи составлялась система из линейных уравнений, где число линейных уравнений равно сумме чисел вспомогательных зарядов электродов и зарядов диэлектрика. Решение этой системы позволяет найти величины всех вспомогательных зарядов, обеспечивающих выполнение граничных условий на поверхностях катода камеры, поверхностях сигнальных провололок и на поверхности диэлектрической рамки камеры.

В первую группу системы входили уравнения, обеспечивающие выполнение граничных условий для нормальных компонентов электрического поля на границе раздела двух сред $E_{n1} = \varepsilon E_{n2}$.

Вторая группа уравнений связана с граничными условиями на поверхности диэлектрика для потенциала: отсутствие разрыва потенциала при переходе через границу раздела между средами 1 и 2 : $\phi_1 = \phi_2$.

Третья группа уравнений связана с обеспечением граничных условий на поверхностях проводников пропорциональной камеры: потенциала катодных плоскостей и потенциала сигнальных провололок.

ФИЗИЧЕСКАЯ МОДЕЛЬ МППК

Схематическое изображение камеры показано на рис. 1. Ее геометрические характеристики: расстояние между катодными плоскостями составляло 10 мм, диаметр сигнальных проволочек равен 20 мкм, боковая рамка камеры имела в сечении форму прямоугольника размером $20,2 \times 10 \text{ мм}^2$. При решении задачи принималось, что величина диэлектрической проницаемости рамки $\epsilon = 4$, сигнальные проволочки находятся под напряжением 3 кВ, а катодные плоскости находятся под нулевым напряжением.

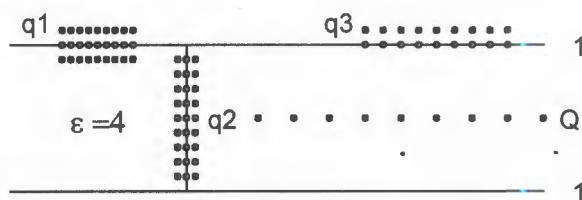


Рис. 1. Схема размещения вспомогательных зарядов пропорциональной камеры. Q — заряды сигнальных проволочек, $q1$ — вспомогательные внешние и внутренние заряды диэлектрической рамки камеры вдоль катодной плоскости, $q2$ — вспомогательные заряды рамки в объеме камеры, $q3$ — вспомогательные заряды катодной плоскости. Контрольные точки, в которых проверяется выполнение граничных условий, размещены на поверхностях катода и рамки против соответствующих вспомогательных зарядов.

При моделировании использовалась следующая система координат: ось абсцисс располагалась в плоскости сигнальных проволочек, ось ординат — по нормали к плоскости проволочек, ось аппликат — вдоль проволочек. Началом координат служила точка на внешней поверхности рамки. Камера состояла всего из 20 проволочек, этого числа вполне достаточно для решения поставленных задач [1]. Расстояние между сигнальными проволочками равнялось 2 мм.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МППК

Решение задачи с диэлектрической рамкой проводилось с использованием вспомогательных зарядов, обеспечивающих выполнение граничных условий на поверхностях катода, сигнальных проволочек и диэлектрической рамки [2,3]. Использовались вспомогательные линейные заряды. В соответствии с результатами [1] численное моделирование проводилось на плоскости $Z = 0$. В этой плоскости потенциал и величина электрического поля, создаваемого линейным зарядом, определяются следующими соотношениями:

$$U = \frac{2\sigma}{4\pi\epsilon_0} \ln \frac{L + \sqrt{L^2 + r^2}}{r} \quad \text{и} \quad E = \frac{2\sigma L}{4\pi\epsilon_0 r \sqrt{L^2 + r^2}}.$$

Здесь $2L$ — длина линейного заряда, σ — линейная плотность заряда, ϵ_0 — диэлектрическая постоянная.

Схема размещения вспомогательных зарядов показана на рис. 1. Внешние и внутренние вспомогательные заряды диэлектрика (q_1) в плоскостях, параллельных плоскости $Y = 0$, были размещены с интервалом в 0,1 мм. Их число вдоль одной из стороны рамки равно 404. Интервал между зарядами диэлектрика (q_2) на плоскостях, параллельных плоскости $X = 0$, брался равным 0,05 мм. Общее число этих зарядов равнялось 196. Вспомогательные заряды катодной плоскости (q_3) располагались на расстоянии друг от друга в 0,4 мм. На длине, равной 60 мм, использовался 151 линейный заряд. Из решения линейной системы 771 независимого уравнения находились величины 771 вспомогательного заряда, в том числе 20 зарядов сигнальных проволок. На основании принципа суперпозиции потенциалы и электрические поля вычислялись как сумма полей и потенциалов, создаваемых 1522 вспомогательными зарядами. Линии поля и эквипотенциали на последующих рисунках строились с помощью замены отрезка кривой отрезком касательной [4].

РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Распределение вспомогательных зарядов иллюстрирует рис. 2. Распределение величин зарядов по проволочкам приведено на рис. 2а. График 1 соответствует равенству всех диаметров проволок. Первоначально все 20 проволок имели равные диаметры по 20 мкм. Увеличение диаметров двух первых проволок до 35 и 30 мкм привело к

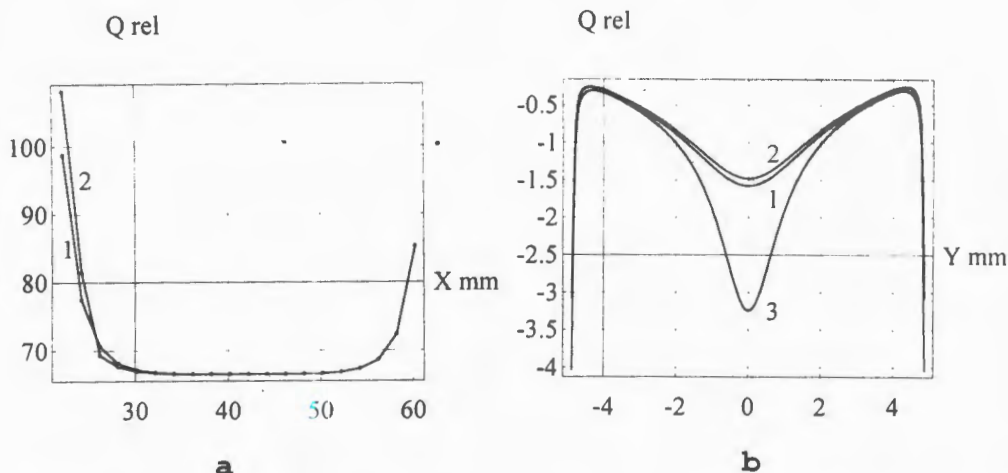


Рис. 2. а) Распределения зарядов сигнальных проволок. Случаю равенства диаметров всех проволок соответствует распределение 1. Распределение 2 относится к случаю увеличения диаметров первой и второй проволок до 35 и 30 мкм. Увеличение диаметров приводит к росту емкости проволок и увеличению их зарядов. б) Распределения вспомогательных зарядов вдоль поверхности диэлектрика. Кривая 1 представляет распределение для проволок равных диаметров. Кривая 2 относится к случаю увеличения диаметров первой и второй проволок. Первая проволока в обоих случаях расположена на расстоянии 2 мм от рамки. Кривая 3 — распределение вспомогательных зарядов, для случая смещения всех проволок к рамке на 1 мм.

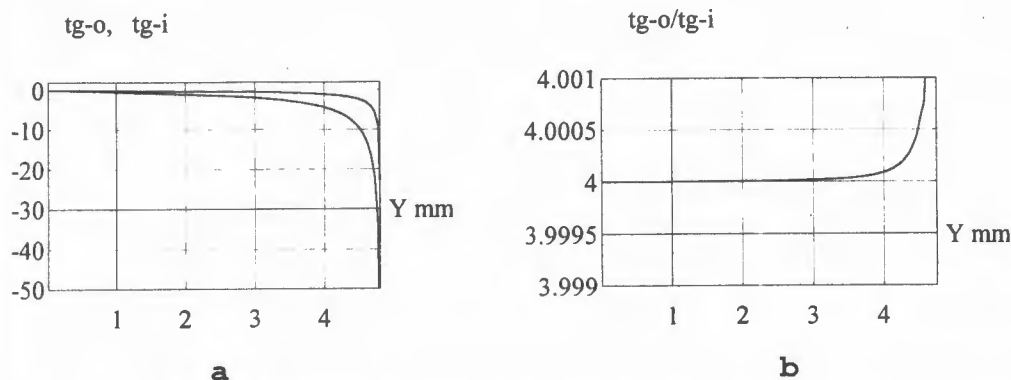


Рис. 3. а) Распределение значений тангенсов углов между линиями поля и нормальными к поверхности рамки камеры. Верхняя кривая относится к углам падения, нижняя — к углам преломления силовых линий поля на границе с диэлектриком. б) Распределение характеризует степень выполнения закона преломления силовых линий $\text{tg } 0 / \text{tg } i = \epsilon$. При решении задачи было задано значение $\epsilon = 4$.

увеличению их линейной емкости. Следствием этого стало увеличение заряда примерно на 10% (график 2).

На рис. 26 показано распределение величин вспомогательных зарядов диэлектрика в объеме камеры. Кривая 1 относится к случаю равенства диаметров всех проволок. Кривая 2 соответствует случаю увеличенных диаметров двух первых проволок. Сдвигу всех проволок к рамке на 1 мм, т.е. на половину расстояния между проволоками, соответствует кривая 3.

Силовые линии на поверхности диэлектрика испытывают излом. Излом характеризуется законом преломления линий поля, по которому отношение тангенсов углов между касательной к линии поля и нормалью с внешней и внутренней стороны диэлектрика должно быть равно диэлектрической проницаемости. Полученные данные величин вспомогательных зарядов позволяют вычислить отношение тангенсов углов падения и преломления. Это отношение характеризует степень достоверности получаемых результатов и точности всех расчетных величин полей и потенциалов.

Распределения величин тангенсов углов падения и преломления вдоль поверхности диэлектрика показаны на рис. 3а. Их отношения приведены на рис. 3б. Кривые на этом рисунке не включают, только последнюю контрольную точку, в которой отношение равно 4,05. Причиной такого отклонения является эффект прямого угла между поверхностью катодной плоскости и поверхностью диэлектрика. В этой точке угол падения составляет около 88° , т.е. угол между линией поля и поверхностью диэлектрика близок к 2° . Значение тангенса угла от 88 до 90° резко изменяется в пределах $30 - \infty$.

Зоны чувствительности первых четырех проволок и эквипотенциали показаны на рис. 4. Диаметры проволок равны между собой. Границами зон являются силовые линии, начинающиеся на проволоках, для которых угол между плоскостью проволок и линиями поля близок к нулевой величине. Наибольшая деформация зон чувствительности приходится на зоны первой и второй проволок. Первая проволока располагается на

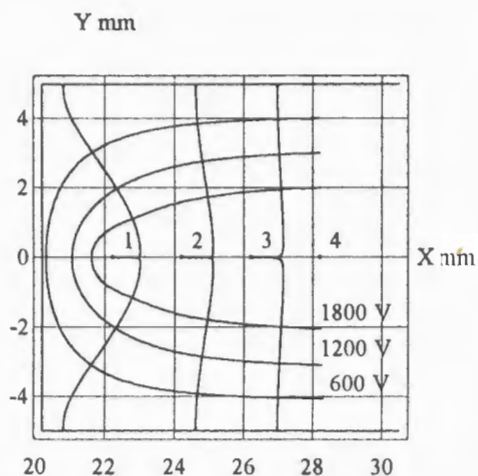


Рис. 4. Зоны чувствительности трех первых проволочек камеры. Вместо прямоугольной формы, свойственной проволочкам центральной области камеры, зоны деформированы. Наибольшая деформация приходится на две первые проволочки. Показанные на рисунке эквипотенциали построены для нулевого напряжения на катоде и напряжения на проволочках, равного 3 кВ.

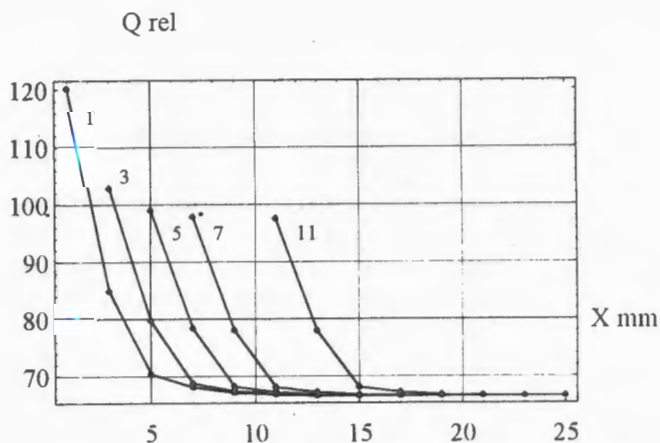


Рис. 5. Распределения зарядов на проволочках камеры для пяти интервалов расстояния между первой проволочкой и рамкой камеры. Эти интервалы указаны на рисунке в мм. Начиная с интервала в 5 мм заряды соответствующих проволочек изменяются мало: влияние диэлектрической рамки становится минимальным.

расстоянии 2 мм от поверхности рамки. С увеличением расстояния между первой проволочкой и рамкой влияние диэлектрика уменьшается. Для нахождения оптимального расстояния между первой проволочкой и рамкой были выполнены расчеты при сдвиге всех проволочек от рамки. На рис. 5 представлены распределения зарядов на проволочках для нескольких интервалов между первой проволочкой и диэлектрической рамкой.

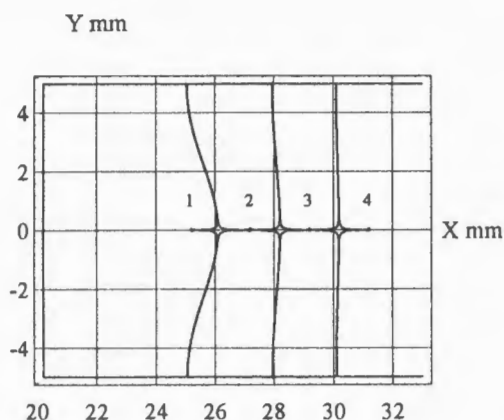


Рис. 6. Зоны чувствительности проволок камеры при интервале между первой проволокой и катодом, равном 5 мм. Эти зоны по сравнению с зонами при интервале первая проволока — катод, равном 2 мм (рис. 4), деформированы в меньшей степени. Зона третьей проволоки является практически прямоугольной. Эта проволока может выполнять функцию первой рабочей для съема информации.

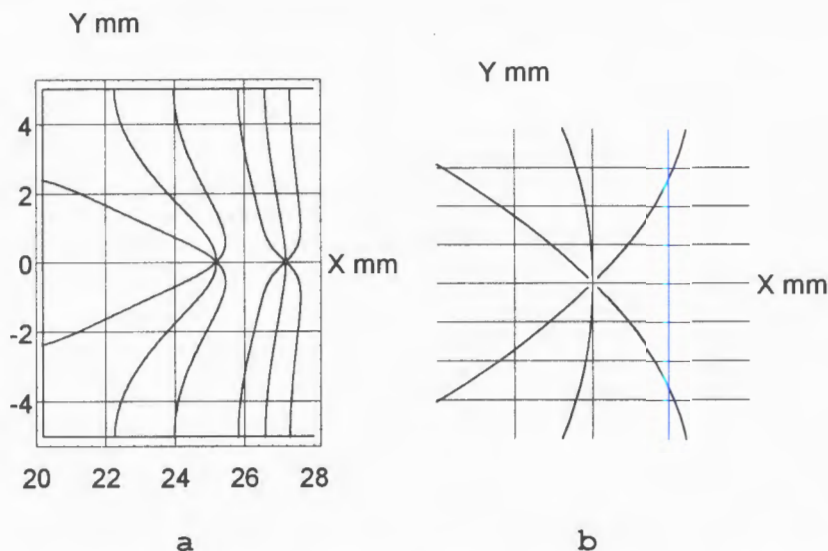


Рис. 7. а) Линии электрического поля двух первых проволок камеры при интервале между первой проволокой и катодом, равном 5 мм. Линии соответствуют углам между плоскостью проволок и вектором поля, равным 45° , 90° и 135° . б) Линии поля в окрестности первой проволоки в увеличенном масштабе.

При величине интервала от $s = 5$ мм заряды проволок практически не меняются. Напряженности электрического поля на поверхности первой и второй проволок с учетом их диаметров соответственно равны $0,85E_0$ и $0,8E_0$. Величине E_0 соответствует

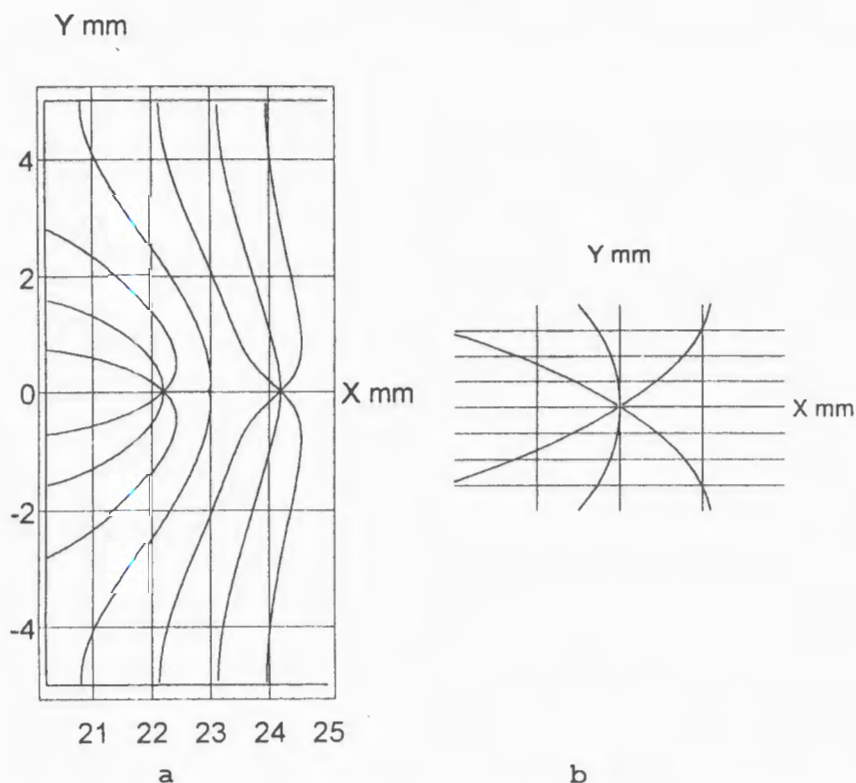


Рис. 8. а) Линии электрического поля двух первых проволок камеры при интервале между первой проволокой и катодом, равном 2 мм. Линии соответствуют углам, равным 0° , 45° , 90° и 135° , между вектором поля и плоскостью проволок. Ни одна из этих линий, кроме первой, не заканчивается на катоде из-за значительного влияния диэлектрической рамки. б) Область в окрестности первой проволоки в увеличенном масштабе.

напряженность поля на поверхности проволок центральной зоны камеры. Максимальная величина вспомогательных зарядов диэлектрика на его поверхности при значении $s = 5$ мм меньше, чем для случая $s = 2$ мм. При расстоянии первой проволоки от рамки в 2 мм в распределении при $Y = 0$ максимальная величина отрицательных зарядов в произвольных единицах равна 1,5 (рис.26). Для $s = 5$ эта величина в три раза меньше.

Форма ячеек первых четырех проволок и границы их зон чувствительности при расстоянии первой проволоки от поверхности рамки $s = 5$ мм показаны на рис. 6. На рис. 7а для интервала $s = 5$ мм показан ход линий поля, начинающихся на поверхности проволок под углом к поверхности проволок в 45° , 90° и 135° . В этом случае только линия поля, выходящая под углом в 135° , заканчивается на поверхности рамки. Для сравнения на рис. 8а показан ход линий поля при расстоянии от первой проволоки до рамки в 2 мм. Для этого случая только линия поля, выходящая под нулевым углом, не заканчивается на поверхности рамки. Сравнение областей в окрестности сигнальных проволок, которые изображены на рис. 7б и 8б, говорит о существовании незначи-

тельной разницы между ними. Это объясняется доминирующим влиянием зарядов самих проволок в их окрестности.

ВЫВОДЫ И ЗАКЛЮЧЕНИЕ

Исследование, результаты которого представлены в настоящей работе, является первым практическим применением методов решения задач о диэлектрике в электрическом поле [2,3]. Именно благодаря разработанным методам решения задач с диэлектриками стало возможным выполнение исследований о влиянии диэлектрической рамки на электростатические характеристики МППК. Это исследование позволяет не только качественно объяснить, но более точно составить представление о способах уменьшения краевых эффектов в МППК.

Первые две проволоки увеличенного диаметра выполняют охранную роль. Для того чтобы напряженность электрического поля на их поверхности не превышала напряженность поля рабочих проволок камеры, первая проволока должна размещаться на расстоянии не меньшем, чем расстояние между плоскостью проволок и катодом. При меньшем расстоянии необходимо использовать охранные проволоки большего диаметра. Увеличение расстояния между рамкой и первой проволокой только уменьшает рабочую площадь МППК.

В [1] показано, что увеличение емкости проволок ведет к уменьшению деформации формы рабочих ячеек первых проволок. По этой же причине диэлектрик рамки камеры способствует уменьшению деформации формы крайних ячеек.

Авторы благодарят В.Д. Пешехонова за поддержку проведенного исследования, М.Г. Шафранову — за обсуждение результатов и ряд замечаний.

Литература

1. Шафранов М.Д. — Краткие сообщения ОИЯИ, 1995, № 6[74]-95, Дубна, с.21.
2. Шафранов М.Д. — Сообщение ОИЯИ P11-99-148, Дубна, 1999.
3. Топурия Т.П., Шафранов М.Д. — Сообщение ОИЯИ P11-99-155, Дубна, 1999.
4. Ландау Л.Д., Лифшиц Е.М. — Электродинамика сплошных сред. М.: Наука, 1992, с.57.
5. Шафранов М.Д. и Топурия Т.П. — Сообщение ОИЯИ P11-99-355, Дубна, 1999.
6. Шафранов М.Д. — Сообщение ОИЯИ P13-96-355, Дубна, 1996.

УДК 539.12.162.8

AN OBJECT-ORIENTED FRAMEWORK FOR THE HADRONIC MONTE-CARLO EVENT GENERATORS

N.Amelin¹, M.Komogorov²

We advocate the development of an object-oriented framework for the hadronic Monte-Carlo (MC) event generators. The hadronic MC user and developer requirements are discussed as well as the hadronic model commonalities. It is argued that the development of a framework is in favour of taking into account of model commonalities since common means stable and can be developed only at once. Such framework can provide different possibilities to have user session more convenient and productive, e.g., an easy access and edition of any model parameter, substitution of model components by the alternative model components without changing the code, customized output, which offers either full information about history of generated event or specific information about reaction final state, etc. Such framework can indeed increase the productivity of a hadronic model developer, particularly, due to the formalization of hadronic model component structure and model component collaborations. The framework based on the component approach opens a way to organize a library of the hadronic model components, which can be considered as the pool of hadronic model building blocks. Basic features, code structure and working examples of the first framework version for the hadronic MC models, which has been built as the starting point, are shortly explained.

The investigation has been performed at the Laboratory of High Energies, JINR.

Объектно-ориентированный фреймворк для создания и использования генераторов событий взаимодействия частиц и ядер

Н.Амелин, М.Комогоров

Обсуждается разработка пакета программ (объектно-ориентированный фреймворк) для создания и использования генераторов событий взаимодействий частиц и ядер. В частности, рассматриваются требования, возникающие при разработке и использовании моделей такого типа, общие свойства присущие моделям данного типа, а также их реализация посредством современной компонентной технологии программирования. Такой подход дает пользователю, работающему посредством единого интерфейса, ряд удобств, например: редактирование любых параметров

¹Nikolai.Ameline@cern.ch

²max@sunhe.jinr.ru

модели, замещение одних компонент другими, конфигурирование выходных данных, которые содержат либо полную информацию об истории генерации события, либо только специфическую информацию о конечном состоянии взаимодействия. Компонентный подход существенно увеличивает производительность разработчика моделей, так как все составляющие пакета реализованы в виде набора формализованных фреймов, описывающих различные типы компонент. Он позволяет организовать хорошо структурированную библиотеку таких компонент с удобной системой навигации по ним, что обеспечивает универсальный и гибкий механизм построения сложных многокомпонентных моделей. В данной публикации также представлены основные составляющие первой версии пакета, приведены примеры его использования. Эта версия может быть рассмотрена как отправная точка для построения более сложной и мощной системы.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

1. INTRODUCTION

The object-oriented approach based on the C++ can be adopted to write the hadronic MC event generators codes. Such approach has many advantages as compared with traditional procedural coding (see [1]). Even more we advocate to build an object-oriented framework [2], since we have observed many commonalities for the hadronic MC models as well as for their usage and for their development. Framework approach is more justified, when the list of models chosen for development is very large and potentially can be increased.

A framework [2] can be considered as well-documented thematic collection of software to build related applications. It outlines the main architecture for the application to be developed. The successful framework should not only support needed features and provide default implementation and built-in functionality as much as possible, but it should also allow an easy modification and extension of the built-in functionality. The main goal of any framework is a reusability. The software developer should be able to reuse written code (e.g., classes from the framework libraries) and the design of a framework. A framework design is closely connected with design patterns used to document certain elements of it. A design pattern is a concise definition of a technique that demonstrates some successful solution for particular coding problem. Particularly, the factories and the proxies design patterns (see book [3]) have been applied in our framework version.

We assume that our object-oriented framework for the hadronic MC generators will be useful for two categories: the model users and the model developers (the advanced users). We consider a hadronic model user as a person who interacts with the framework by means of a user interface without writing and modifying of the model codes. A hadronic model developer is assumed to work with the framework on the level of internal framework interfaces. A developer needs the knowledge of the framework structure and libraries as well as the knowledge of C++ language.

The main idea of our framework is to adopt the component approach. We have in mind that such framework can be the base to build an extending library of the model components and the model algorithms. It can allow us to extract model components from the model components library pool and to compose them into the powerful physical models.

Below we would like to outline the hadronic model commonalities as well as the requirements for hadronic model package from the hadronic model user and developer. Then we

want to explain our framework basic ideas and shortly describe the possible user and developer sessions. To understand the framework performance there are several important parts of its architecture that need to be explained as well as interaction of these parts. Thus we would like to provide short explanation of the model components and their structures, the framework control subsystem (the dispatcher) and its work, the set of classes for the parameter and input maps, the data transfer class library, the output subsystem with the data exchange format and the data file structure, etc. Finally, we offer several examples to illustrate the framework work.

2. COMMONALITIES OF THE MC HADRONIC MODELS

The MC hadronic models can be used as the hadronic event generators with the main goal to study hadronic collision phenomena as well as the source of information about hadronic collision final states with the aim to utilize this information.

Even taking a fast look at the MC hadronic models one can see that they have much in common. First of all they are phenomenological models having large amount of model parameters. We can specify these parameters as the physical parameters (hadronic model tuning constants) and hadronic model configurators. The first type of parameters gives a possibility of changing hadronic model results. They operate similarly as the physical input. This type of parameters fulfils very important job to store physical information about hadronic processes. The second type of parameters also offers a possibility of changing hadronic model results, but by changing the application logic of a physical algorithm.

Additionally to the parameters much more information should be provided for any hadronic model. For example, the information about physical properties of particles: quark contents, electric charges, masses, decay branchings, etc., is required. The information about the physical properties of stable and excited nuclei: binding energies, spins, level density parameters, fission barrier heights, etc., is also required. Usually such information is needed in the read-only mode.

The MC hadronic models act in a similar way. They either convert an input into an output or they use an input to give an answer to the user request. The input can be the characteristics of particles: hadrons, partons, gammas, etc., or characteristics of nuclei: stable nuclei, excited fragments, etc. The output can be again a set of particles: hadrons, partons, gammas, etc., or a set of nuclei: stable nuclei, excited fragments, etc. Acting so any MC hadronic model has to deal with four vectors (energy-momentum and time-position) and their transformations. Any hadronic model somehow handles the n -body kinematics.

The most of hadronic models are the multicomponent models. A multicomponent model includes other models as additional or alternative model components and has complicated execution flow. Especially for application purposes a user needs the set of hadronic models to obtain proper description of the hadronic reaction final states [4].

Practically all hadronic models are complicated numerical models. For them it is not always trivial to separate «physics» from «algorithm», i.e., to separate the physical input, physical parameters etc., influence on the simulation results from the chosen numerical algorithm influences. It is also often, when the same numerical algorithm can be reused within models describing different physical phenomena, e.g., the decay of resonances and the excited nuclei according to the relativistic phase space, the elastic scattering of partons,

hadrons and nuclei, the search collision and decay algorithm for the hadron transport model and the parton transport model, etc.

The different kinds of errors can be occurred during hadronic model initialization or model runtime. The source of errors can be due to the inconsistent user input as well as due to the complexity of numerical algorithms. The last situation is an often unexpected situation.

Any hadronic model is required to produce different physical outputs, which should be analyzed. The output can be only specific information about hadronic reaction final states or complete information about the history of a generated event.

The above list of commonalities can be more extended. For example, besides kinematics all hadronic MC models deal with random sampling of variables according to the different probability distributions, i.e., large set of the random number generators are required. To perform numerical operations, different mathematical utilities: equation solvers, integrators, interpolators, etc., are needed to be employed. However, it is already clear that a hadronic model developer should take into account these commonalities by either common code structure or common used methods or common implementations, etc.

3. TYPICAL REQUIREMENTS FROM HADRONIC MODEL USER AND DEVELOPER

Any user of the hadronic models needs model descriptions, i.e., the reaction mechanisms, model inputs, model parameters, used numerical algorithms, etc., should be described. Such descriptions should be accompanied by needed examples to demonstrate model usage.

The different usage strategies provide different user requirements for hadronic model package. A user performing theoretical or experimental study of hadronic collision needs a possibility to «play» with chosen model, e.g., the possibility to visualize and change model parameters, to configure model or model component, to choose an alternative model component, to customize hadronic model output. Thus, the mechanism to check consistency of the user alterations should be provided. For this type of user, run control requires to have hadronic model runtime information and exception handling mechanism.

Another type of user or applied user are mostly interested in the generated event itself. The configured for a given type of hadronic reaction model with default values of parameters should be offered for the applied user. The output information should be reduced until required minimum and presented in the required form.

Of course, both types of users need to have much more, e.g., simple and self-explanatory mechanism for hadronic reaction input preventing from errors due to the inconsistent input, the mechanism to throw out event due to the possible severe errors at runtime, the analysis and visualisation tools are also required to analyze the generated output, etc. Thus, for any kind of users a user session should be convenient and productive.

The model developers may want to rebuild an existing hadronic model with aims to extend the range of its applicability, to increase its predictive power, etc. They may want to incorporate an existing («foreign») hadronic model for a cooperative work with other existing models. Also they may want to build a new hadronic model running standalone or in collaboration with other models. The enumerated situations are the primer tasks of a hadronic model developer. But in reality to satisfy the user requirements he or she should realize the user interfaces and even do much more. The interfaces between created hadronic model code

and the outside world, e.g., an operational system, are also needed to be realized as well as different adapters, if he or she wants to use an external packages, e.g., to visualize and analyze the data. A developer should have in mind its hadronic model package portability. The package portability means not only the portability in the sense to work on different computer platforms, but also, it is the possibility for hadronic package to work within another package. A hadronic model developer should facilitate the future developers tasks as well.

In the conventional approach for the developing of the hadronic MC models a developer or several developers are working independently on a particular model. Such approach, even if an object-oriented language is used, has several drawbacks. First of all the hadronic model commonalities as well as designing and programming experiences are badly exploited. For example, each new developer has usually started to develop a particular hadronic model from scratch. As a result of it, each new model developer starts from the analysis and design stages. The design duplications are manpower consuming. The different designs have different qualities and they provide different degrees of satisfaction for the user as well as the future model developers requirements. The design duplications lead also to the code duplications. The particular hadronic model code quality depends strongly on the coding experience of model developer or developers. It also becomes more difficult to learn, maintain and extend a set of hadronic models created by different developers as well as to connect them for collaborative work. As a rule, the hadronic model developers are not the software experts, they are physicists and experts in their subject domains. For them it can be difficult to find a proper solution of the specific software tasks.

Thus, we advocate the development of an object-oriented framework to take into account hadronic model commonalities, to facilitate user work and to increase productivity of the developer work. Within it a hadronic model developer can be more deeply concentrated on the particular model problems. He or she needs to write much less code since an essential part of the program already exists. He or she does not need to be a software expert to write robust code. A new model code inherited from the framework could also be much easily tested since it is already integrated with the rest of framework.

4. FRAMEWORK BASIC IDEAS

As was already discussed, the hadronic models as well as their usage and their development have a lot of commonalities. We tried to separate commonality from variability in the hadronic model interfaces and application logics during the framework development.

Here we begin to describe the first version of the hadronic MC model framework, which is suggested as the starting point. Let us explain the framework basic ideas and its collaboration with users and model developers.

We made an attempt to formalize hadronic model components as unit blocks to construct a composite hadronic model. All such blocks can be stored as an extendible collection of the model components. The definition and design of the unit component blocks offer a universal, flexible and powerful mechanism for a hadronic model construction. The question is how to define such components? Any model component can be structured into an interface part and the part presenting its application logic. The interface part of the model component allows component collaborations. With the help of this part a user can also handle model component

parameters, its input-output, its execution process, etc. Such formalization becomes visible, when we provide the component interface standards.

Particularly, from model components we can separate the components with the «runnable» interfaces, i.e., they include methods to execute them and support user input. We call a model component as the main component if it uses other model components and supports the runnable interface.

On the other hand, the interface standards, if they are required, dictate a hadronic model developer to follow definite rules during a component implementation. These rules can be taken into account by means of the component frames, which are created for the command line user interface, or by means of the component wizards in the case of the graphical user interface. The component wizard helps us to create a particular case of a component. We call it the empty or skeleton component. A developer can also create a skeleton using the component frames and performing the required editions. Thus, the interface standards facilitate the developing of the model component interface part.

A hadronic model developer should mostly work on the implementation of a particular application logic. The part, which presents an application logic can also be universal. It is known that the same application logic can be used in several physical models. Below we will touch more details of the application logic universality.

In the case of developing a composite hadronic model there are very important questions about the component control, their collaboration and their collaboration with a user. Particularly, a component collaboration should allow substitution of variable implementation of different application logics (e.g., the alternative components) via a common interface. Again the interface standards help us to suggest mechanism for model component control and their collaboration. This mechanism is provided by the framework control subsystem. The control subsystem is implemented by means of two central concepts of the framework: the resource and the dispatcher. The resource contains static information about model components. The dispatcher is the heart of our framework. It loads components, obtains information about components, creates needed files, and so on. The dispatcher is the chief of all components. It controls all inner model processes at runtime. But any model component is controlled by the dispatcher in the same way by means of the set of standard messages. The dispatcher helps us to organize multithread work of our framework. The dispatcher concept allows an easy way to integrate our framework into the «more global» package, e.g., the *GEANT4* [1] or the *ROOT* [5].

Our framework is a tool to perform simulation of hadronic interactions. Therefore, we have suggested a uniform and powerful output subsystem with main goal to facilitate simulated data visualisation and analysis. This output subsystem is based on the extendible data transfer class library.

The data transfer class library as well as the application logics of the model components are closely connected with the problem domain. The objects of the data transfer classes are used for an information exchange between hadronic model algorithms. These classes help us to store the history of an event generation and to implement the universal model algorithms. Again the data transfer class library helps a hadronic model developer to be concentrated on the application logics development.

At the end of this chapter we want to say a few words about the object identifiers. Any object of our framework has its unique identifier (*ID*). At the moment any *ID* can have the unsigned value, thus we have a possibility to assign more than 4 billion different values. The

unsigned values are very convenient for searching and navigation. The knowledge of object's *ID* helps us to obtain full information about this object. The object identifiers are heavily used in our framework subsystems. Particularly, we are developing the framework help subsystem based on the *HTML*. It is possible due to that the unique object identifiers can be binded with the *HTML* files describing objects.

5. POSSIBLE USER AND DEVELOPER SESSIONS

For communications with user we have developed the user interfaces. Here we explain the user-framework interaction mainly by means of the command lines. However, a user using the Windows platform can run the graphical user interface.

First of all a user is able to visualise the hadronic model component list and choose or register needed component. The *HTML* based help subsystem under development will offer a user needed model documentation. After model component registration a user can visualise and edit the default model parameters and the framework will provide parameter consistency check. The visualised set of the hadronic model parameters can have a tree structure with the leaves are groups of parameters. The hadronic model parameter set can have three states: default state, current state and the state, which includes one or several previous states stored in a file. Thus a user can either keep a default value of a parameter variable or change this value. Also a possibility for user to store current value of parameter variable on a file is offered. It can be done in order to give the «undo» possibility, i.e., a user will always be able to get back the old parameter values.

A hadronic model can be configured by its parameter edition. Even more advanced possibility for a user to configure a composite model is offered. He or she is able to substitute a model component by an alternative model component without changing the code (see the component substitution chapter).

Then a user can customize output. Similarly as a parameter set, the output set can be a tree, where leaves are channels and branches are groups of channels. Each channel or group of channels can be in the enabling or disabling states. By default they are forced to be in the disabling states. Using this scheme we offer a possibility for a user to control the output information. For example, preparing an output of a particle object we can obtain either full information about a particle, i.e., its momentum, its position, its encodings, its spin, its electric and baryon charges, etc., or only its momentum and its position. Also in the case of a multicomponent model to obtain the history of the generated event a user can activate any hadronic model component or all hadronic model components to write information about its work.

The input set is organized similarly as the parameter set since both the input data and the parameters require that their values should be checked. For each particular model the input maps (see an example of input maps in chapter 17) are offered and a user will be able to edit them and again the framework will check the input consistency.

For particular hadronic model a user is able to overload default aggregated component set, default parameter set, default output configuration. It is useful, e.g., in the case, when a hadronic model was tuned for the best description of experimental data and a user is interested only in generated final states. Such situation is typical if a model is used for the applied purposes to predict missing information about hadronic reactions.

The hadronic model run control is offered by the model runtime information: information messages, warning messages and error messages.

Since the framework has uniform output subsystem it helps us to create the generated data analysis and visualisation subsystem. Now simple data filters and some plot facilities are provided for the user, who uses the graphical user interface. We should note that within the graphical user interface the user-framework collaboration is similar as in the case of the command line user interface. It is going through the parameter, input, output, runtime information, etc., graphical windows. In this case a user deals with several threads, e.g., the run process or active model component thread and the data analysis thread. In the command line user interface, a user operates with the corresponding ASCII files.

Another possibility of performing more detailed analysis of simulated events is the use of an external analysis tools. The output subsystem is able to prepare the output data to be suitable as the input (macros) for the *ROOT* [5] analysis and visualise tool.

For efficient work a developer can learn the set of framework classes (see also chapters below): **NMAlgorithm** class as the starting point to implement different model components, several classes, such as the **NMDoubleParam**, the **NMIntParam**, etc., to handle model parameters and inputs as well as **NMHistory**, **NMBaseParticle** and **NMPrimerParticle** classes to extend the data transfer class library if it is needed. For other framework classes a developer can learn only their headers. It is also better if a developer makes the acquaintance with the hadronic application logic library to avoid a universal algorithm duplication. The implementations of some classes are closely connected with the interface system, e.g., command line or graphical user interface. Their implementation should be hidden from a model developer since we may want to change them without affecting the developer's codes.

To help a model developer we provide several model components frames as well component documentation frames. An example of such frame can be found in the appendix chapter. If a developer wants to implement a new model or a new model component, e.g., as an alternative model component, he or she can choose suitable model component frame, edit it according to the frame comments and implement required methods. The component documentation frames help a developer to create the help documentation about a component.

The simplest case of a developer's work is that a developer wants to implement foreign working code as an alternative model component. Particularly, to implement foreign runnable component code a developer has to use the runnable algorithm frame, where only two methods: *OnRegisterInputMap()* and *OnRun()* (see the appendix chapter) should be implemented. A developer has also to define the input map and map contained variables and register them by *OnRegisterInputMap()*.

6. MODEL COMPONENTS AND THEIR STRUCTURES

We distinguish different kinds of model components: the algorithm component, data analysis component, table component, etc. Each type of component can have its specific properties and methods. Such approach has some advantages for a model developer, since we can provide the standard frames for all components to facilitate component implementations. It is also rather flexible to allow the implementation of complicated model [6]. From the hadronic model developer point of view the most important model component is the algorithm

component. Let us explain the details of its structure. Below we will call it shortly the model component or the component.

For this component we separate the component factory and the component static information parts. We need the component factory part, because the creation of objects is very important question since we want to have a possibility for dynamic binding, i.e., substitution of a component by the alternative component without changing the code as well as object creation only, when it is needed. Another aspect of this question is connected with model component object collaboration (see below). There are two ways, when one object uses another object. In the first case we can declare it as a member and we cannot substitute it during run time. In the second case we declare a pointer to another object and we will get a runtime substitution. Thus we can substitute this object by its child or derived object. But we have to assign this pointer. As a rule it can be done by a set method or using a constructor. It is not always good because we may want to use a complex object and we need to know in aggregating object all information about aggregated object. To avoid such problem we can create the proxy object [3] of an object, which is planned to be used. It can be a pointer to the component identifier (ID). During construction time all proxy objects are gathering into a list. As a result, each object knows, which aggregated objects can be used. But how can we create these aggregated objects? It can be done by means of a factory method [3].

The component static information part or component resource part contains the information, which is needed to create this component. It also contains the pointers (references) to documentation. Besides the component factory and its static information parts, a component can include methods and data for model parameters, model input maps and model output configurations as well as the execution or run methods.

To provide more flexibility for a developer, we also classify the algorithm components according to their interfaces: the so-called «runnable» components, which include the execution and input methods. Such components can be used as the separate models or as the main model components, which include aggregated components. We distinguish also the «general» components, which can be used only as the aggregated components. We should note that any runnable component can be used as a general component. We classify the «virtual» components, i.e., the «virtual runnable» or «virtual general» components, which are similar to the «runnable» and «general» components. The difference between them is in the component factories. It is not possible to create the virtual component objects.

At the end of this chapter we would note that the **NMAlgorithm** class is the base class to derive the algorithm component classes (see the appendix chapter), which should be implemented. Particularly, it supports three methods: *OnOverloadDefaultProxy()* to overload the default component, the *OnOverloadDefaultParam()* to overload default parameters and *OnOverloadDefaultOutputConfig()* to overload default output configuration.

7. FRAMEWORK CONTROL SUBSYSTEM

Let us imagine that all needed model components are implemented. How is the user's interaction with the framework libraries provided? How will components collaborate? As we have already mentioned, all interactions between the framework libraries and user interface system as well as model component collaborations are provided by the framework control subsystem. It is hidden from a user. We denote such control subsystem as the dispatcher.

Thus, the dispatcher is a shell between the user interface and the framework libraries. For the command line interface, the framework control subsystem is currently implemented as the **NMDispatcher** class, which keeps the pointer to the model component base class as well as the vector of the model component factories. It includes many methods (see chapter 17) to support several phases of the user-framework interaction: creation phase, edition phase, execution phase and destruction phase. Particularly, it allows one to obtain full list of the model components included into the framework. The dispatcher object can load, run and delete a model component. The dispatcher object checks a possibility of substituting a component by an alternative component. The dispatcher object also manages parameter and input writing and reading. It manages writing and reading of the output configurations, etc. There are two groups of methods in the dispatcher. The first group consists from the methods needed to receive the information about framework library contents and to control this information, e.g., the method to obtain the component tree from the global list. The second group are the methods, which control currently loaded components. The dispatcher methods are named according to the rule: *Action + ComponentType()*, e.g., *LoadAlgorithm()*, *LoadAnalyzer()*, *LoadTable()*, *DeleteAlgorithm()*, etc., since we have different model components.

8. PARAMETERS AND INPUT SUBSYSTEMS

Our framework has the set of classes to support model parameter and model input handlings. A developer to implement the parameters of his model component has to define parameter type, assign its default value and a comment for it. Provided model component frames facilitate this task for a developer. After model component development a model user will get a possibility to edit parameters under the framework control. A developer can also bound parameter values by means of some parameter method overloading.

To solve the task of the input data set we suggest the input maps based on the lists of simple data types. By means of the input maps we organize not only the uniform input for a model user, but we also save developer time since our framework performs needed input map data converting.

The **NMParam** class initiates the parameter/input class branch. This class keeps common properties of parameters, e.g., parameter names, methods to write parameters into data file, to read parameters for their visualization, to check parameter consistency, to reset parameters, etc. The **NMParam** class gives rise to the **NMParam** group of classes: **HM***Param** and **NMParamGroup**. The **NMParam** realizes common features of a simple parameter. The necessity of such a class is connected with difference between writing into file for a simple parameter and for a group of parameters independent from parameter type. The **NM***Param** presents parameters of different types. The stars in the parameter class name indicate that any name can be used according to the rule: **Bool + NM***Param = NMBoolParam**. The **NMParamGroup** class allows us to join parameters into named group.

As we already mentioned the parameters are structured like tree and have different states. A model component can contain only one group of parameters. Thus, the parameters tree is the component tree, where leaves are groups of parameters. The parameter tree is created by the proxy component tree. There are default state and current state. Also several former states can be stored in a configuration file. If the configuration file is not empty, the last state of a parameter tree will be loaded from this file.

Similar structure is also suitable for model input. To support input maps we use the `NMInputMap` class, which is a redefinition of the `NMParameterGroup` class.

9. MODEL COMPONENT SUBSTITUTION

Our framework approach allows substitution of model components by the alternative components without changing code. It gives a user the possibility to change hadronic model structures. Two types of the substitution are possible. The first one we call the static substitution. Let us imagine the next situation. There are several model components that have been developed. We denote them as *A*, *B*, *C* and *C1*. The component *A* uses the component *B* and the component *B* uses the component *C*. When a developer creates these components, he or she knows nothing about the component *C1*, which is an alternative for the component *C*. A user applying the method `OnOverloadDefaultProxy()` in the component *A* can force the component *B* to use the component *C1*. The implementation of the `OnOverloadDefaultProxy()` method can look as follows:

```
ClassA::OnOverloadDefaultProxy()
{
    OverloadProxy(ID_C1, ID_B, 0)
}
```

The dynamic substitution can be performed at run stage. For example, a user would like to use the component *C* instead of the component *C1*. He or she is able to solve this task by means of our framework. Within the command line interface version it can be done by edition of the parameter file, which has the `.nmp` extension (see also chapter 17). The parameter file edition includes the replacement of the lines related to the component *C1* by the lines related to the component *C*. From the C++ programmer point of view such substitution is provided by the subclassing mechanism, i.e., alternative components need to have a common base class. The developed component frames of our framework supports such subclassing.

10. DATA TRANSFER CLASS LIBRARY

The data transfer classes organize the data transfer between model components. This is the first goal to develop such library. It helps us to store the history of an event generation. Any object of the data transfer class supports serialisation, i.e., it has methods to read and write its state. The data transfer classes use the exchange data format described below. Thus, the data transfer class library helps us to create uniform and powerful output subsystem to obtain the result of simulation. For this purpose any data transfer class is derived from the `NMHistory` class with the aim to obtain an output of the generated event history. It has two methods `Config()` and `Out()` that are implemented to write their objects states. (See examples of the method implementations in the framework output subsystem chapter). This is the second motivation to create such hierarchical library.

We would stress that the presence of these classes allows a developer to pay more attention to the algorithmic logic part of a hadronic model development, because a developer need not think about details of the input-output operations.

A navigation system can be offered (we are working on it) and a developer can navigate through the data transfer classes. He or she can visualise any such class and apply the data transfer class wizard with the aim to extend the data transfer class library.

The data transfer classes are tightly connected with the hadronic models domain. The data transfer class hierarchy represents physical objects from this application area. They describe three groups of such objects. The first one can be called the simplest objects or the objects without inner structure. Then we can separate collections of the simplest objects. Describing the simplest object we do not need the information about its inner structure. We have developed the **NMBaseParticle** class. It has two members: the 4-momentum and the 4-position, and represents the relativistic phase space point object or relativistically moving point object. This class can be considered as an example of the data transfer class, which represents the simplest object. Other examples of such classes, which are derived from the **NMBaseParticle** class, are the **NMParticle** class, representing the elementary particles, the **NMParton** class, representing the quarks and gluons, and the **NMSimpleNucleus** class, representing a nucleus without its nucleon structure. The simplest objects can be combined into the different object collections. The members of a collection do not have structural dependences. For example, object collections can be created using the **NMBaseParticleVect**, the **NMParticleVect**, the **NMPartonVect** and the **NMSimpleNucleusVect** classes. They keep arrays of the pointers to the simplest objects. The composite objects need the information about their internal structure. The classes, representing such objects, have the collection classes as the members or they are derived from the collection classes. As the examples of this group of classes we can consider the **NMNucleus** class, which has as a member the **NMParticleVect** class, and the **NMExitedString** class, which has the **NMPartonVect** class as a member.

For many applications of any class hierarchy it is very convenient to have a base class, which joins all classes. As a rule, it is pure virtual class. It factorizes the common properties of all objects of this hierarchy. The relativistic phase space point object is commonly used in the high energy hadronic MC models. In our data transfer class hierarchy we use for that the **NMPrimeParticle** class, which represents the most operations needed to work with the relativistic phase space point. This class gives the rise to all other classes of the hierarchy.

Our data transfer classes represent not only the dynamical properties of the physical objects (momenta, positions, etc.), but also their static properties (charges, encodings, etc.). For example, the **NMParticle** object describes the dynamic properties of the elementary particle and has a pointer to the static object, which keeps the information about mass, decay branchings spin, etc. The static information is stored as tables. We have other tables, e.g., the interaction cross section table. We are working on the service tools (framework database service tools) to support tables within our framework.

At the end of this chapter we want to mention another important motivation to create the hierarchial library of the data transfer classes. It is connected with the algorithm logic universality. The degree of universality of an algorithm is determined by the amount of needed input-output information. Our data transfer classes have an hierarchial structure, the most universal algorithms are those, which use as input and output the objects of the node data transfer class.

11. FRAMEWORK OUTPUT SUBSYSTEM

The output phase is the most complicated phase of interaction between the framework and a user. Let us make several things to be explained before description of the output phase.

There are different approaches to write and read data. One of them is that an object writes and reads data by itself. In this case we need this object to be created before reading. The configuration to be read is provided by this object class and we need not store the configuration on a file. But in such a case the data stored on a file cannot be read from outside of a package, which includes the object class. We have decided to use another approach. We write the configuration to be read at the beginning of a file. This configuration is based on the list of basic types. These types can be read within any other package.

We have developed the framework input-output file system to avoid dependence on the different platforms. Particularly, it allows us to open and close a file, to read data from a file and to write data on a file.

Let us consider the output data structure, which can be written on an output file. It includes the output file header, an extra information, the channel definition, the event configuration and the data part.

The output file header is needed to identify the output file. It contains the identifier field, the file version and some offsets to the other parts of file as the channel definition, the event configuration and the data. An extra information, which can be used by a developer for some reasons, will be placed just after the header.

We need a universal data exchange mechanism, which is independent of particular language or package. For this purpose we suggest a format for data reading and writing. A variable is stored in a channel with its own identifier. With each channel identifier we bind the channel type and some extra information to organize convenient user interface. Several channels can be grouped into a group of channels with unique group identifier. Each identifier of channel or group of channels can have prefix *REP*. It signals that given channel or given group of channels is repeatedly stored. To describe a channel we use four fields: *ChannelID*, *FullName*, *ShortName*, and *UnitName*. The *ChannelID* field keeps information about the channel type and it can also be used in the help subsystem. The channel full name *FullName* field is used by the user interface. The *ShortName* field can simply keep the short name of a channel. For example, the *X* component of the Lorentz vector momentum can be presented as follows: 1003, «X-component of Lorentz vector», «px», «GeV». The *ShortName* field was also introduced to be a variable name of this channel. Using this field we can customize the output data structure with the aim to produce output, which can be accepted as an input for the foreign package. For example, it helps us to create the *ROOT* [5] macros. The last field *UnitName* informs a user about the physical units of channel variable values. To support channels we have developed the *NMChannelDef* class and its vector *NMChannelDefVect* class.

We have introduced program event objects. We define a program event object as an unit of information to be written. To support it we have developed the *NMProgramEvent* class. Each program event object has its own *EventID* and can have its name to be used in the user interface. In order to read the binary data from the output file we have to know their configuration. The configuration can be found by means of the *EventID* in the configuration part of the file. From this point of view the *EventID* can be considered as an identifier of the event object type.

The configuration is array of *ChannelID*, which are included in this event. Each model algorithm component supports the method *OnRegisterOutputConfiguration()*. We use this method to register configurations of program events, which can be written by this component. The registration procedure is very easy. It adds the channel identifiers or group of channel identifiers to the array. An example of the method *OnRegisterOutputConfiguration()* implementation can be found in chapter 17.

We said already that the data exchange between the model components is supported by the data transfer classes. These classes play an important role in the output subsystem. In order to solve the task of writing event configuration to the output file all these classes have to realize the method *Config()*. We present below two examples of this method implementations.

```
void NMBaseParticle::Config(NMProgramEvent& ProgramEvent, bool bHistory)
{
    ProgramEvent.AddGroup(bHistory, 10, true, "Moving point",
    "MovePoint");
    ProgramEvent.AddDouble(101, true, "X-position", "X",
    "fermi");
    ProgramEvent.AddDouble(102, true, "Y-position", "Y",
    "fermi");
    ProgramEvent.AddDouble(103, true, "Z-position", "Z",
    "fermi");
    ProgramEvent.AddDouble(104, true, "T-position", "T",
    "fermi");
    ProgramEvent.AddDouble(105, true, "X - component of
    Lorentz momentum", "PX", "GeV");

    ProgramEvent.AddDouble(106, true, "Y - component of
    Lorentz momentum", "PY", "GeV");
    ProgramEvent.AddDouble(107, true, "Z - component of
    Lorentz momentum", "PZ", "GeV");
    ProgramEvent.AddDouble(108, true, "Energy", "E", "GeV");
    ProgramEvent.AddEndGroup();
}
```

```
void NMParticle::Config(NMProgramEvent& ProgramEvent, bool bHistory)
{
    ProgramEvent.AddGroup(bHistory, 30, true, "Particle");
    NMBaseParticle::Config(ProgramEvent);
    ProgramEvent.AddDWord(301, true, "Encoding", "Enc",
    "Encoding");
    ProgramEvent.AddDouble(302, true, "Mass", "M", "GeV");
    ProgramEvent.AddDouble(303, true, "Charge", "M");
    ProgramEvent.AddEndGroup();
}
```


The data transfer class objects do not exist, when we create the output configuration, as a result, these methods have to be the static methods.

We define event configurations before run session. It gives us a possibility to control the output. Particularly, we are able to protect any channel or group of channels from their output. By default all channels are open and all program events are closed for their output except the program events from the main model component.

The framework supports also two predefined event objects, which allow us to write to output the values of input map and parameter of model components, which are executed in the current run session. For them the registration of channels is performed automatically. However, a user is able to mask these events or some their channels with the aim to forbid their output.

There are different *Out()* methods, which are responsible to write the data part of the output file. We should note that before to write data on the output file each such method checks that there are channels, which open for the output. As we already mentioned each data transfer class has such method. For example, for the *NMBaseParticle* and the *NMParticle* it has been implemented as follows:

```
void NMBaseParticle::Out(NMOutputConfig& Config, NMFile& File)
{
    BEGIN_OUT_GROUP
    {
        OUT(X());
        OUT(Y());
        OUT(Z());
        OUT(T());
        OUT(PX());
        OUT(PY());
        OUT(PZ());
        OUT(E());
    }
    END_OUT_GROUP;
}
```

```
void NMParticle::Out(NMOutputConfig& Config, NMFile& File)
{
    BEGIN_OUT_GROUP
    {
        NMBaseParticle::Out(Config, File);
        OUT(Encoding());
        OUT(Mass());
        OUT(Charge());
    }
    END_OUT_GROUP;
}
```

In the above *Out()* method example the *OUT(Variable)* is the macro to write a *Variable*.

Let us say a few words about the output of the generated physical event history organization using as an example the **NMProgramEvent** class. It has also the *Out()* method. The *Out()* method of **NMProgramEvent** class writes on output file the header (*EventID*, *EventCounter*, and *HistoryCounter*) and then binary data. Each call of the *Out()* method can be considered as the event object creation and it can be identified using the *EventCounter*. There will be no event objects written on the output file with the same value of the *EventCounter*. The *HistoryCounter* field is used to make connection between event objects and model component objects, which are responsible for the event object creation. The *HistoryCounter* variable is the only a member of the **NMHistory** class. With the aim to obtain the output of the generated physical event history, we have also derived all data transfer classes from the **NMHistory** class. But in some cases an event object can contain not only objects of the data transfer classes. There will also be event objects, which do not contain any object of the data transfer classes. If so, we can also write within events the values of the *HistoryCounter* as a member.

At the end of this chapter we want to discuss the generated data analysis procedure, which can be performed within the graphical user interface of our framework. We consider this procedure as the three-steps procedure. At the first step one can perform the structure filtering since the data are organized as a tree, where nodes are enumerated by their *ID*. It gives us a possibility to select data by using only the data structure information, which is stored at the beginning of data file. The next step is to perform the data sensitive filtering, i.e., one can select data using themselves. The last step is that selected data can be stored and presented as the one-dimensional histogram and two-dimensional plots.

12. RUNTIME INFORMATION

Our framework supports the runtime information output. By means of it the model components inform about their execution processes during run session. The runtime information is simple text based information. There are three types of the runtime messages: information messages, warning messages, and error messages. The information messages are used to tell about normal execution process. The warning messages inform a user about potential errors or other situations, which are able to destroy normal execution process. The execution process will be continued after the appearance of the warning messages. The error messages appear, when the execution process will be aborted. The error messages inform also a user about place and type of the error. The framework detects itself the information about place of an error and a developer need not make special efforts to solve this task.

A user has a possibility to control the runtime information output. Particularly, he or she can either totally suppress the information messages or suppress the information messages from some model component objects. The warning messages are divided into some groups. A user can suppress any group to be appeared in the output or overload any group to produce error messages. Each warning or error message has also its own unique *WarningID* or *ErrorID* identifier. It allows us to help a user or a developer to obtain the full information about the reason of warning or error.

Let us provide more details about the possibility of obtaining the runtime information. We have developed the **NMOwner** component base class, which includes the methods: *Message()*,

`Warning()`, and `Error()` to support runtime messages. The error method calls also exception to close the execution process. All objects of the model component classes have pointers to their context objects. The work to organize the runtime output is performed by the context object of a model component. The dispatcher creates the context objects and informs them, where the runtime information has to be placed.

13. FRAMEWORK HELP AND DOCUMENTATION

The help subsystem based on the *HTML* is under development as well as printed version of the framework manual [6]. Such subsystem is needed to help a user and a potential hadronic model developer. They can obtain parameter description, input description, physics description, numerical algorithm description, etc. The help subsystem from a hadronic model developer point of view can be considered as a platform to explain how to develop a model code.

As we already said, any object of our framework such as component, parameter, error message, etc., has the unique *ID*. It allows us to bind these *ID* with the object describing *HTML* files. For version with the graphical user interface we will provide some tools to work with the help information, e.g., there is a possibility of obtaining the context of help information. This help subsystem will be developed with the aim to provide either convenient access for help information or uniform way to add new help information by means of typical documentation frames. The user of help subsystem need not know the details of its work in order to find or add help information. The component documentation frames allow one to present information for the standard appearance. These documentation frames are different for different types of model components. They are developed in the complement of model component frames. A model component developer must follow the instructions, which can be found in the documentation frames. There are also classes such as the `NMDeveloper` and the `NMDeveloperList`, which organize the personal information about model developers, and the `NMWEBDocument` and the `NMWEBDocumentList` classes, which assist to manage the model documentation on the *Web*. See also the example of runnable component frame in chapter 17.

14. COMPONENT LIBRARY

Discussing different aspects of the hadronic model framework, we have paid attention mostly to its interfaces. But the real hadronic model component to be used for the reaction simulation should have its application logic. The hadronic model algorithm transforms an input into an output. The objects of the data transfer classes are served as the input and output objects for the model component objects. In the runnable model component frame the `OnRun()` method name was chosen as the standard name for the transformation method. Thus, we assume that the implementation of this method will be the main job of a hadronic model developer. It usually requires hard developer's work and needs not only the software testing, but also the physical testing. The large number of the hadronic model runnable components, e.g., such as the `NMElasticScatterer` and the `NMPomeronPartonStringModel` (see for model physics description [4]) as well as hadronic model virtual and general components have been

already implemented and included into the component library. The component library will be described elsewhere [6].

15. UTILITY CLASS LIBRARY

The utility classes contain useful data and methods: random number generators, different integrators, equation solvers, physical units and constants, etc., which are often used. These classes accumulate the working experiences of the hadronic model code programmers. Some of such utility classes can also be found in the *CLHEP* library [7]. Very convenient strategy to use physical units and constants was adopted from the *GEANT4* [1].

16. CONCLUSIONS

We have shortly discussed the first version of the object-oriented framework for the hadronic MC event generators. We made an attempt to explain that our framework has different features to facilitate the hadronic model user and developer works. As the framework name we suggest to use the *NiMax* word.

In conclusions, we would only stress the main point of our approach to develop the hadronic MC event generators. It is the component approach, when a complex hadronic model is composed of small and simple pieces, that are self-contained entities (see, e.g., the book [8]). The component approach has many advantages and we can enumerate several of them. It allows us to formalize the particular type of components by separating interface part and application logic. We have offered different component frames for hadronic model developers. Thus, a model developer should work on the component application logics and each component application logic can be developed independently of other component application logics. Model components can be composed in a variety of ways and the new components with their peculiarities can be added, that offer a flexibility for the construction of a powerful hadronic model. The different implementations of a component application logic can be interchanged at runtime enabling a hadronic model user to obtain the needed model properties without redesigning a model and writing the model code.

Working on the framework we have obtained good software experience and we understand that first framework version should be more tested, improved and extended to be the successful tool for its user and consider it as the starting point. We hope also that our experience to develop the hadronic model framework as well as the framework ideas can be applied not only in the hadronic model application domain, but can be suitable to develop other similar applications.

Finally, we would like to thank the members of the *GEANT4* hadronic group for the discussions with respect to the hadronic MC generator design. One of us (N. Amelin) is grateful for S. Giani, who invited him to work on the hadronic models for the *GEANT4* toolkit. Our MC generator framework idea was born during this work. M. Komogorov would like to express special thanks to W. Trzaska for collaboration during his work at Physics Department of the Jyväskylä University.

17. FRAMEWORK WORKING EXAMPLES

With the aim to illustrate possible framework user and developer sessions we show several examples: The first is an example of the framework main function, which can be applied in the case of the command line user interface. This example explains the dispatcher importance. Then comes the example of the runnable algorithm component frame illustrating a user and a developer works with the parameters, the input maps and the output configurations. We show also the example, when a user is able to substitute a model component by the alternative model component. Finally, we would demonstrate an example of the heavy ion collision simulation performed within the Pomeron based Parton String Model (see the model physics description [4]).

17.1. Dispatcher at Work. Let us look at the first example. As a result of this main function execution a user can obtain the list of the model components (see file «Content.hm») as well as the information about registered (the model ID has been assigned) model parameters («.nmp» file), model input maps («.nmi» file) and model output configuration («.nmo» file). The corresponding files can be edited. Model user can execute registered model and obtain the configured output, i.e., the result of the hadronic reaction simulation, which will be written on an output file («.nmd»). In this example of the main function one can see the key role of the dispatcher. It is doing all control job, which is hidden from a user.

```
#include <iostream.h>
#include "NM.h"
#define MAX_BUFF 256

void main(int cArg, char* aArg[])
{
    Dispatcher.RegisterFactory();
    if (cArg < 2)
    {
        cout << "To get help, please, use:\n"
"<NiMax -I> - list of components\n"
"<NiMax -I ID [-iMapFName] [-pParamFName] [-oOutFName]>\n"
"  Subkey:\n"
"  -i - Input maps\n"
"  -p - Parameter list\n"
"  -o - Output configuration tree\n"
"To run a component, please, use:\n"
"<NiMax -R ID [-iMapFName] [-pParamFName] [-oOutFName] [DataFileName]>\n"
"  -i - Input map file\n"
"  -p - Parameter set file\n"
"  -o - Output configuration tree file\n"
" DataFileName - Output file name\n";
        return;
    }
    if (strcmp(aArg[1], "-I") == 0)
    {
```

```

if (cArg < 3)
{
Dispatcher.OutComponentsList("Content.nm");
return;
}
int ID = atoi(aArg[2]);
if (!Dispatcher.LoadModel(ID))
{
cout<<"There is no component with ID = "<< ID << endl;
return;
}
if (cArg < 4)
{
char Buff[MAX_BUFF];
itoa(ID, Buff, 10);
strcat(Buff, ".nmi");
Dispatcher.InputToTextStream(Buff);

itoa(ID, Buff, 10);
strcat(Buff, ".nmp");
Dispatcher.ParamToTextStream(Buff);

itoa(ID, Buff, 10);
strcat(Buff, ".nmo");
Dispatcher.OutputConfigToTextStream(Buff);
return;
}
for(int c1 = 3; c1 < cArg && c1 < 6; c1++)
{
if (strnicmp(aArg[c1], "-i", 2) == 0)
Dispatcher.InputToTextStream(aArg[c1] + 2);
if (strnicmp(aArg[c1], "-p", 2) == 0)
Dispatcher.ParamToTextStream(aArg[c1] + 2);
if (strnicmp(aArg[c1], "-o", 2) == 0)
Dispatcher.OutputConfigToTextStream(aArg[c1] + 2);
}
}
if (strcmp(aArg[1], "-R") == 0)
{
if (cArg < 3)
{
Dispatcher.OutComponentsList("Content.hm");
cout<<"\nPlease! Set the ID\n";
return;
}
int ID = atoi(aArg[2]);

```



```

if (!Dispatcher.LoadModel(ID))
{
    cout<<"There is no component with ID = "<< ID << endl;
    return;
}
for(int c1 = 3; c1 < cArg && c1 < 6; c1++)
{
    if (strnicmp(aArg[c1], "-i", 2) == 0)
        Dispatcher.InputFromTextStream(aArg[c1] + 2);
    if (strnicmp(aArg[c1], "-p", 2) == 0)
        Dispatcher.ParamFromTextStream(aArg[c1] + 2);
    if (strnicmp(aArg[c1], "-o", 2) == 0)
        Dispatcher.OutputConfigFromTextStream(aArg[c1] + 2);
}

    Dispatcher.Run("kyky.nmd");
}
}

```

17.2. Runnable Model Component Frame. As we already said, a model developer to create a model component can start the component implementation by editing the suitable model component frame. For example, the runnable algorithm component can be used. Its header and implementation are shown below. We shall assume that all members are public to simplify the discussion; the extension to protected and private is not straightforward. We plan to explain it in the extended description [6] of our framework. Let us at first explain the parameter, input and output parts of the runnable model component. Below we consider implemented parts of runnable component for the hadronic string decay model. This model performs string decay simulation, i.e., it models the process, when a string with the parton ends and its mass decays into stable and resonance hadrons (see the model physics description in [4]).

17.2.1. How to Define, Edit and Check the Model Component Parameters? In order to define parameter one has to declare the member of component class. There are corresponding commented lines in the runnable component frame (see below). Let us consider a part of the **NMStringDecayer** model component header. Within it we declare two double parameters: *DiquarkSuppression* and *ClusterMass*.

```

class NMStringDecayer : public NMAgorithm
{
public:
    DEFAULT_LOCAL_DEFINITION(NMStringDecayer, NMAgorithm);

public:
    // Messages

public:
    // Proxy

```

```

public:
    // Parameters
    NMDoubleParam ClusterMass;
    NMDoubleParam DiquarkSuppression;
public:
    // Input

public:
    // ProgramEvent

public:
    // Methods

};

```

The framework is able to control these parameters, if they are registered by the *OnRegisterParam()* method. Below we show an example of the method implementation.

```

void NMStringDecayer::OnRegisterParam()
{
    Add(ClusterMass.Init(200908, "Cluster mass", 0.15, GeV, "GeV"));
    Add(StrangeSuppress.Init(200910, "Strangeness suppression", 0.44));
}

```

Within the command line user interface the component parameters can be written on the file with the extension *.nmp* and read from it. Below we show the content of *.nmp* file for the registered parameters from the above examples.

```

"NMStringDecayer", 2009
{
    "Cluster mass", "GeV", 200908,          0.15;
    "Strangeness suppression", "", 200910,    0.44;
}

```

We can edit this file, e.g.

```

"NMStringDecayer", 2009
{
    "Cluster mass", "GeV", 200908,          0.18;
    "Strangeness suppression", "", 200910,    0.40;
}

```

This file can be used as the parameter of the framework execution file. Let us call it as the *NiMax.exe*. Also framework gives a possibility to check edited parameter consistency. For this purpose a developer can overload the *OnCheck()* method, e.g.,

```

bool NMDoubleMinMaxParam:: OnCheck()
{
    if (Value < Min || Value > Max)
    {
        Warning(123,ERR_GROUP,"Parameter value is out of range. Default value
will be used");
        return false;
    }
    return true;
}

```

17.2.2. How to Define, Edit and Check the Model Component Input Maps? The definition of an input map is similar to the parameter definition, besides that, one has at first to declare input map. One can find the corresponding comment about it in the example of the runnable component frame (see below). Let us consider an example of the input map *InputMap1* declaration for the string decay component. This map has three fields: *LeftQuarkEncoding*, *RightQuarkEncoding*, and *StringMass*.

```

class NMStringDecayer : public NMAgorithm
{
public:
    DEFAULT_LOCAL_DEFINITION(NMStringDecayer, NMAgorithm);

public:
    // Messages

public:
    // Proxy

public:
    // Parameters
    NMDoubleParam ClusterMass;
    NMDoubleParam DiquarkSuppression;

public:
    // Input
    NMInputMap InputMap1;

    NMIntParam LeftQuarkEnc;
    NMIntParam RightQuarkEnc;
    NMDoubleParam StringMass;

public:
    // Methods

};

```


Similar to the parameters, an input map and its content have to be registered. There is the method *OnRegisteredInputMap()* for that. Below we show an example of this method.

```
NMStringDecayer:: OnRegisterInputMaps()
{
    NMAgorithm:: OnRegisterInputMaps();
    AddMap(InputMap1.Init(2009100, "Input map 1 for string decayer "));
    InputMap1.Add(LeftQuarkEnc.Init(2009101,"Left quark encoding", 2,
"Encoding"));
    InputMap1.Add(RightQuarkEnc.Init(2009102,"Right quark encoding", 2201,
"Encoding"));
    InputMap1.Add(StringMass.Init(2009103,"String mass", 200, GeV, "GeV"));
}
```

In command line user interface version this map can be written on the file with the extension *.nmi* and read from it. Let us look at the content of the *.nmi* file as a result of the input map registration, which was considered above.

```
"Input map 1 for string decayer ", 2009100
{
    " Left quark encoding ", "Encoding", 2009101, 2;
    " Left quark encoding ", "Encoding", 2009102, 2212;
    " String mass ", "GeV", 2009103, 200;
}
```

It is possible to offer different input maps for one component. For example, we can add the string momentum for string decay. To do that we have to add the declaration of new map and new input map values.

When somebody uses the input maps, then a map from the set of maps should be registered as the default map by the *OnRegisterInputMap()* method. If it is not done, then the framework will consider the first registered input map as the default one.

There can be some dependences between input or parameter values. For example, on string ends should be quarks with the opposite colours. To solve this problem, each runnable component supports methods *OnCheckParam()* and *OnCheckInput()*. An example of the *OnCheckInput()* implementation is shown below.

```
NMStringDecayer:: OnCheckInput(NM_ID ID)
{
    if (IsColor(LeftQuarkEnc) && IsAntiColor(Right QuarkEnc) ||
        IsAntiColor(LeftQuarkEnc) && IsColor(Right QuarkEnc))
        return NMAgorithm:: OnCheckInput(ID);
    Warning(127, ERR_GROUP, "String end quarks have the same colours");
    return false;
}
```

17.2.3. How to Define and Edit the Output Configurations? We have already discussed that the output is performed by means of program events. Each event has to be declared. There is appropriate comment in the runnable component frame.

In the example, which is shown below, we add two program events for the **NMStringDecayer** component named *InitialState* and *FinalState*.

```
class NMStringDecayer : public NMAgorithm
{
public:
    DEFAULT_LOCAL_DEFINITION(NMStringDecayer, NMAgorithm);

public:
    // Messages

public:
    // Proxy

public:
    // Parameters
    NMDoubleParam ClusterMass;
    NMDoubleParam DiquarkSuppression;
public:
    // Output
    // ProgramEvent

    NMProgramEvent InitialState;
    NMProgramEvent FinalState;

public:
    // Input
    NMInputMap InputMap1;

    NMIntParam LeftQuarkEnc;
    NMIntParam RightQuarkEnc;
    NMDoubleParam StringMass;

public:
    // Methods
};
```

After declaration of the program events, they have to be registered. Within the command line version of the user interface the registered output configuration will appear in the *.nmo* files. Below we show the part of this file for the **NMStringDecayer** example.

```
"NMStringDecayer", 2009
{
    "Initial state", 20090001, Off
    {
        "Particle", 20, On
```

```

{
  "Moving particle", "MovingParticle", 10, On
  {
    "X-position", "X", "fermi", 101, On;
    "Y-position", "Y", "fermi", 102, On;
    "Z-position", "Z", "fermi", 103, On;
    "T-position", "T", "fermi", 104, On;
    "X - component of Lorentz momentum", "PX", "GeV", 105, On;
    "Y - component of Lorentz momentum", "PY", "GeV", 106, On;
    "Z - component of Lorentz momentum", "PZ", "GeV", 107, On;
    "Energy", "E", "GeV", 108, On;
  }
  "Encoding", "Enc", "Encoding", 301, On;
}
"Particle", 20, On
{
  "Moving particle", "MovingParticle", 10, On
  {
    "X-position", "X", "fermi", 101, On;
    "Y-position", "Y", "fermi", 102, On;
    "Z-position", "Z", "fermi", 103, On;
    "T-position", "T", "fermi", 104, On;
    "X - component of Lorentz momentum", "PX", "GeV", 105, On;
    "Y - component of Lorentz momentum", "PY", "GeV", 106, On;
    "Z - component of Lorentz momentum", "PZ", "GeV", 107, On;
    "Energy", "E", "GeV", 108, On;
  }
  "Encoding", "Enc", "Encoding", 301, On;
}
"String Mass", "Ecms", "GeV", 2009555, On;
}
"Final state", 20090002, On
{
  "Particle", 20, On
  {
    "Moving particle", "MovingParticle", 10, On
    {
      "X-position", "X", "fermi", 101, On;
      "Y-position", "Y", "fermi", 102, On;
      "Z-position", "Z", "fermi", 103, On;
      "T-position", "T", "fermi", 104, On;
      "X - component of Lorentz momentum", "PX", "GeV", 105, On;
      "Y - component of Lorentz momentum", "PY", "GeV", 106, On;
      "Z - component of Lorentz momentum", "PZ", "GeV", 107, On;
      "Energy", "E", "GeV", 108, On;
    }
  }
}

```



```

        "Encoding", "Enc", "Encoding", 301, On;
    }
}

```

Using the *On/Off* words one can open or close a channel to write it on the output file.

Finally, at the end of this subchapter we show an example of the runnable model component.

```

#ifndef _ClassName_
#define _ClassName_ 1

#include <NM.h>
// #include "ClassParentName.h" // Edit this line, if you use subclassing.

// Here, include the header files for each used child component.
// #include "ChildComponentName.h"

DEFAULT_GLOBAL_DEFINITION(ClassName, ClassParentName)

class ClassName : public ClassParentName
{
public:
    DEFAULT_LOCAL_DEFINITION(ClassName, ClassParentName);

public:
    // Messages
    virtual void OnOverloadDefaultProxy();
    virtual void OnCreate();
    virtual void OnDestroy();
    virtual void OnRegisterParam();
    virtual void OnOverloadDefaultParam();
    virtual void OnRegisterOutputConfig();
    virtual void OnOverloadDefaultOutputConfig();

    virtual void OnRegisterInputMaps();
    virtual bool OnRun(unsigned, NM_ID);

public:
    // Proxy

    // NMAAlgorithmProxy < ChildComponentClass > ChildComponentVariable;
    // Repeat the previous line for each used child component.

public:
    // Parameters

```

```

// Here, define the parameters.
NMDoubleParam P1;

public:
    //Output
    // ProgramEvent

    //Here, define the program events.
    NMProgramEvent Event1;

public:
    // Input

    //Here, define the input maps.
    NMInputMap Map1;

    NMIntParam      IP11;
    NMIntParam      IP12;
    NMDoubleParam   IP13;

    NMInputMap Map2;

    NMIntParam      IP21;
    NMIntParam      IP22;

public:
    // Methods

    // Your methods and data members.
};

#endif

#include "ClassName.h"

// Here, define name of your component and its own ID.
BEGIN_RUNNABLE_IMPLEMENTATION(ClassName, ClassParentName, ID , "ClassName")
{
    // If you want to provide the full information about yourself,
    // please, use this structure.
    DEVELOPER("Physics design", //Explain your participation in the component development.
        "FirstName SecondName Surname", // Your full name
        "Nickname",                      // Your nickname
        "kyky1@cern.ch",                 // Your e-mail
    );
}

```

```

"www.cern.ch",           // Your WEB home page
"7-096-216-53-18",      // Your phone number
"7-096-216-53-18");     // Your FAX number

// If you do not want to provide the full information about yourself,
// please, use this structure.
Developer.Comment("Software design");
Developer.Name("FirstName SecondName Surname");
Developer.NickName("NickName");
Developer.e_mail("kyky1@cern.ch");
Developer.HomePage("www.cern.ch");
Developer.Phone("7-096-216-53-18");
Developer.FAX("7-096-216-53-18");
Add(Developer);
    // Repeat previous lines for each developer.

// Here, define the HTML pages, where your component is described.
    WEBPAGE("\\MyComponent\\Documentation\\File1.html",
//Path for the HTML file.
    "Physics reference");           // Explain the HTML file content.

    WEBPAGE("\\MyComponent\\Documentation\\File2.html",
    "Software reference");
    // Repeat previous lines for each reference.
}
END_RUNNABLE_IMPLEMENTATION(ClassName, ClassParentName)

// Proxy

void ClassName::OnOverloadDefaultProxy()
{
    ClassParentName::OnOverloadDefaultProxy();
    // Here, you can overload the child components
    // OverloadProxy(<NewComponentID>, <Path>, 0);
    // Path is the list of IDs
}

//Create and destroy

void ClassName::OnCreate()
{
    ClassParentName::OnCreate();
    //Here, load the needed data
}

void ClassName::OnDestroy()

```

```

{
    // Release the data, which were allocated by the previous method.
    ClassParentName::OnDestroy();
}

void ClassName::OnRegisterParam()
{
    ClassParentName::OnRegisterParam();

    // Here, register your component parameters.
    Add(P1.Init(201301, "Name of parameter", 0.45, GeV, "GeV"));
    // Repeat previous line for each parameter.

}

void ClassName::OnOverloadDefaultParam()
{
    ClassParentName::OnOverloadDefaultParam();

    //Here, you can overload default parameter values
    // for child components.
    // OverloadDoubleParam(<NewValue>, <ParameterID>, <Path>, 0);
    // Path is the list of IDs.
}

// Output

void ClassName::OnRegisterOutputConfig()
{
    ClassParentName::OnRegisterOutputConfig();

    AddEvent(Event1, 2013001, "Name of event");
    // NMParticle::Config(Event1);
    // NMParticlePtrVect::Config(Event1);
    // Event1.AddDouble(1300, "Centre mass system energy", "Ecms", "GeV");

}

void ClassName::OnOverloadDefaultOutputConfig()
{
    ClassParentName::OnOverloadDefaultOutputConfig();
}

// Input

```



```

void ClassName::OnRegisterInputMaps()
{
    ClassParentName::OnRegisterInputMaps();

    //Here, register input maps and their contents
    AddMap(Map1.Init(2013100, "Map1 name"));
    Map1.Add(IP11.Init(2013101, "P11 Name", 2212, "Encoding"));
    Map1.Add(IP12.Init(2013102, "P12 Name", 2212, "Encoding"));
    Map1.Add(IP13.Init(2013103, "P13 Name", 200, GeV, "GeV"));

    AddMap(Map2.Init(2013200, "Map2 name"));
    Map1.Add(IP21.Init(2013201, "P21 Name", 2212, "Encoding"));
    Map1.Add(IP22.Init(2013202, "P22 Name", 2212, "Encoding"));
    Map1.Add(IP13); // This variable has already been initialized

    aInputMap.Default = 2013100; // default input map
}

bool ClassName::OnRun(unsigned ParentCounter, NM_ID MapID)
{
    if (MapID == Map1.ID)
    {
        // Your code
        return true;
    }
    if (MapID == Map2.ID)
    {
        // Your code
        return true;
    }
    return ClassParentName::OnRun(ParentEventCounter, MapID);
}

// Your code

```

17.3. How to Substitute a Model Component? Let us assume, that we have created the *NiMax.exe* execution file (see also the main function example). Let us assume, that we have main component with its identifier value 2013 and two child alternative model components, which are registered with the values of their identifiers: 2009 and 2015, respectively. Let us replace the 2009 component by the 2015 component. At first one should run the command: *NiMax -I 2013* to obtain the file *2013.nmp*, where the 2013 component tree with parameters can be found. Then one should run the command: *NiMax -I 2015* and obtain the file *2015.nmp* with the component parameter tree for the 2015 component. Then within an editor one has to replace the parameter branch 2009 in the 2013 component parameter tree by the 2015

tree. After that one should run command: `NiMax.exe -R 2013 -i2013.nmi -p2013.nmp` to get generated physical events.

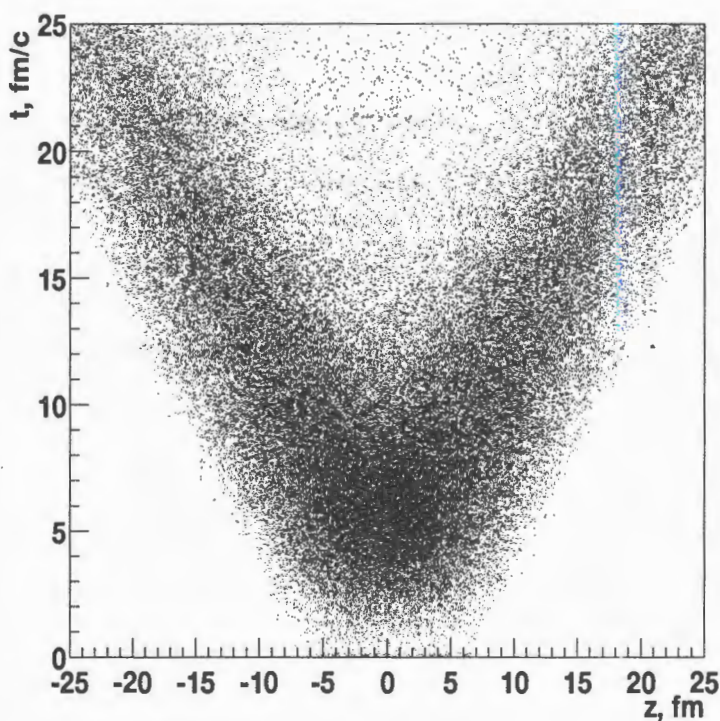


Figure. Space-time distribution of the produced hadrons

17.4. Central Uranium-Uranium Collision Simulation. For the last example we have performed the simulation of central uranium on uranium collisions at the centre of mass energy 6500 GeV per nucleon within the Pomeron based Parton String Model (see physics description in [4]). This model is a complicated model and it was composed of 12 different components.

We have simulated 10 events and created 2-dimensional distribution of the produced particles in space-time (the Z-direction is the beam-direction), which is shown in the Figure. The particle coordinates are calculated at particle freeze-out times. We did not separate the negligible impact of possible spectator nucleons.

References

1. Wenaus T. et al. — GEANT4: An Object-Oriented Toolkit for Simulation in HEP, CERN/LHCC/97-40.

2. Taligent's Guide: Building Object-Oriented Frameworks, [http : //www.ibm.com/java/education/ooibuilding/index.html](http://www.ibm.com/java/education/ooibuilding/index.html).
3. Gamma E., Helm R., Johnson R., Vlissides J. — Design Patterns. Elements of Reusable Object-Oriented Software, Eddison-Wesley, 1995.
4. Amelin N. — Physics and Algorithms of the Hadronic Monte-Carlo Event Generators. Notes for a developer., CERN/IT/99/6.
5. Brun R. et al. — ROOT System, CERN/HP, 1997. See [http : //root.cern.ch/](http://root.cern.ch/).
6. Amelin N., Komogorov M. — NiMax Manual. Component-Oriented Framework for Hadronic MC Generators, in preparation.
7. Class Library for High Energy Physics, [http : //www1.cern.ch/asd/lhc + /clhep/index.html](http://www1.cern.ch/asd/lhc+/clhep/index.html)
8. Orfali R., Harkey D., Edwards J. — The Essential Distributed Objects. Survival Guide., John Wiley and Sons, Inc., 1996.

УДК 621.039.5, 621.384.6

УРАН-238 КАК ИСТОЧНИК ПОЛУЧЕНИЯ ЭНЕРГИИ ЭЛЕКТРОЯДЕРНЫМ МЕТОДОМ

В.П.Дмитриевский¹, Н.В.Сергеева²

Рассмотрена возможность использования изотопа U^{238} для получения энергии в процессе деления его на быстрых нейтронах при использовании внешнего источника нейтронов в подкритической сборке. Физический смысл разработки такой технологии определяется двумя основными проблемами ядерной энергетики — ядерная безопасность и переработка радиоактивных отходов. Первая проблема полностью решается при наличии подкритичности, вторая — частично за счет увеличения в десятки раз (по сравнению с реакторами) срока эксплуатации сборки. Наличие порога деления U^{238} (~ 1 МэВ) определяет необходимость работы на энергетическом спектре нейтронов, который соответствует процессу деления (отсутствие замедлителя). Внешний источник нейтронов создается на базе сильноточного ускорителя. Перечислены физические исследования, необходимые для разработки такой схемы получения энергии.

Работа выполнена в Лаборатории ядерных проблем ОИЯИ.

Uranium-238 as a Source for Electronuclear Power Production

V.P.Dmitrievsky, N.V.Sergeeva

A possibility of using the U^{238} isotope to produce power through fission induced by fast neutrons from an external neutron source in the subcritical assembly is considered. Physically, development of this technology is dictated by two basic problems of nuclear power production, namely, nuclear safety and transmutation of radioactive waste. The former problem is fully solved by using subcriticality, the latter is partly solved by increasing the service life of the assembly by tens-fold (as compared with reactors). The U^{238} fission threshold (some 1 MeV) makes it necessary to work with the neutron energies corresponding to fission (without a moderator). The external neutron source will be based on a high-current accelerator. The necessary physical investigations for development of this power production scheme are listed.

The investigation has been performed at the Laboratory of Nuclear Problems, JINR.

¹e-mail: vpd@nusun.jinr.ru

²e-mail: nata@nusun.jinr.ru

ВВЕДЕНИЕ

Современные ядерные реакторы, в которых используется цепная реакция замедленных нейтронов, работают на изотопах урана-235 и -233, где отсутствует энергетический порог реакции деления и сечение деления возрастает в сотни раз для медленных нейтронов, что приводит к необходимости использования замедлителей нейтронов из легкого вещества (углерод, дейтерий). Успешное развитие ядерной энергетики по этой схеме позволило обеспечить 10–30% получения необходимой электроэнергии в наиболее экономически развитых странах. Этот процесс продолжает развиваться, охватывая все больше стран.

Расширение процесса ставит перед разработчиками ядерных реакторов новые задачи, главная из которых сводится к снижению вероятности возникновения неконтролируемой цепной реакции и захоронению (или переработке) радиоактивных ядерных отходов. Исследования в этих направлениях ведутся во многих физических лабораториях [1–12].

Однако наличие большого запаса естественного урана, а также развитие ускорительной техники, где средние мощности пучков ускоренных частиц достигают мегаваттных мощностей, указывают на целесообразность разработки систем, достаточно удаленных от критичности, для которых накопленные продукты деления в течение длительного периода не оказывают существенного влияния на процесс выделения тепловой энергии при работе на энергетическом спектре нейтронов деления.

Возможность разработки такой технологии по предложению А.М.Балдина рассматривается в настоящем сообщении.

1. ПРОЦЕСС ДЕЛЕНИЯ УРАНА

Естественный уран, получаемый промышленным способом, состоит из смеси двух изотопов U^{238} и U^{235} , причем количество последнего составляет $\sim 0,7\%$. Так как для современной атомной энергетики интерес представляет только изотоп-235, происходит накопление урана-238 в больших количествах, вопрос хранения которого не вызывает особых затруднений из-за его невысокой радиоактивности (период полураспада $4,5 \cdot 10^9$ лет)¹.

Изотоп делится только на быстрых нейтронах. Сечения деления представлены на рис. 1, из которого следует, что порог деления лежит в области энергии нейтронов ~ 1 МэВ ($\sigma_f \leq 10$ мб) в то время как при энергии больше 1 МэВ $\sigma_f = (0,6 \div 1,0)$ б, что соответствует длине свободного пробега в уране по процессу деления

$$\ell_f = 1/(N\sigma_f) = (36,6 - 22) \text{ см}, \quad (1)$$

где N — число ядер в см^3 .

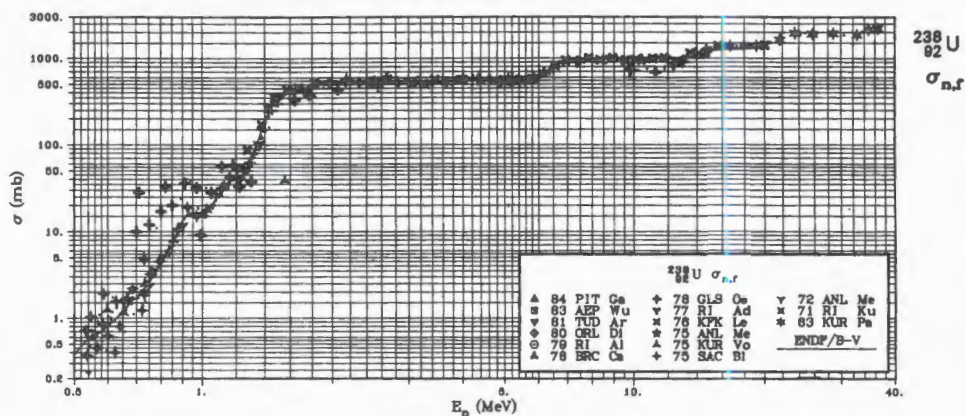
Неупругие сечения, конкурирующие с делением процессов захвата нейтрона, основные из которых $\sigma(n, \gamma)$, соизмеримо или превышает сечение деления, что приводит к поглощению потока моноэнергетических нейтронов в пропорциях

$$\Delta\Phi_f = -\Phi_0 \frac{\sigma_f}{\sigma_3 + \sigma_f}, \quad \Delta\Phi_3 = -\Phi_0 \frac{\sigma_3}{\sigma_3 + \sigma_f}, \quad (2)$$

где σ_3 — сечение захвата нейтрона без деления.

¹ Атомиздат, Справочник, с.913, 1976 г.

Neutron Cross Sections, vol.2, INC, 1988

Рис. 1. Сечение деления U^{238} в зависимости от энергии нейтрона, вызвавшего деление

Из (2) непосредственно следует, что отношение поглощения моноэнергетических нейтронов процессами захвата и деления определяется величиной

$$\alpha = \sigma_3 / \sigma_f, \quad \sigma_3 \geq \sigma_\gamma. \quad (3)$$

На рис. 2,3 приведены значения σ_γ и α в зависимости от энергии нейтрона при $\sigma_3 = \sigma_\gamma$. Так как спектр нейтронов деления описывается распределением

$$f(E) = \frac{2}{\sqrt{\pi T^3}} \sqrt{E} e^{-\frac{E}{T}}, \quad (4)$$

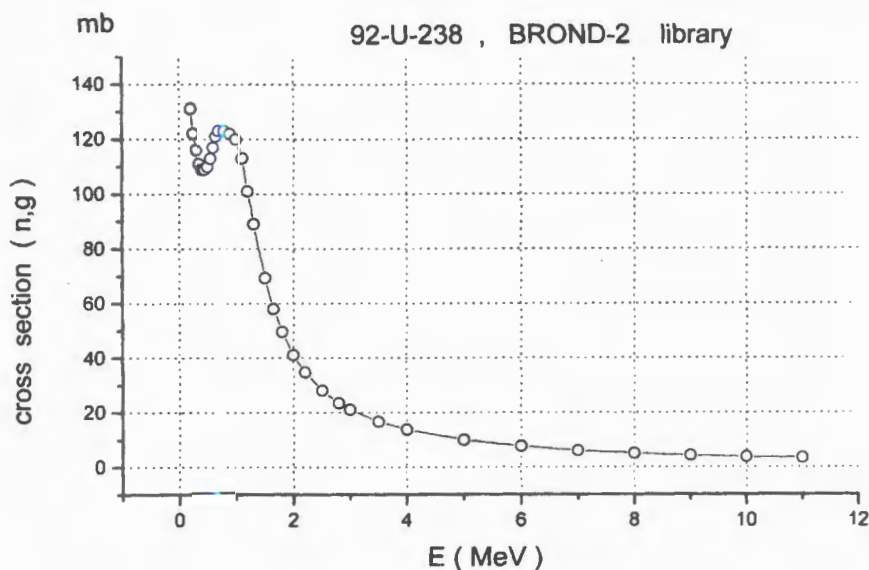
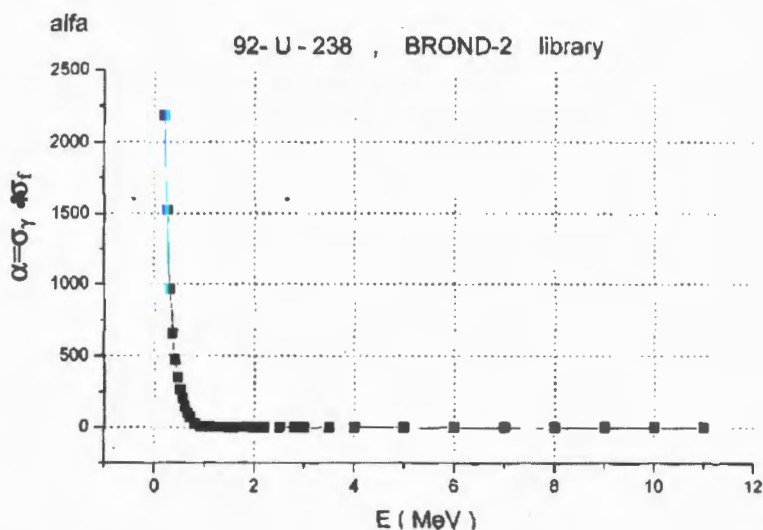
где T — температура ядра при захвате нейтрона, значение α для энергий спектра деления можно вычислить из выражения

$$\alpha_g = \int_0^\infty f(E) \frac{\sigma_\gamma}{\sigma_f} dE. \quad (5)$$

Для урана-238 при высоком энергетическом пороге деления разброс температур при делении невелик. Это позволяет в первом приближении рассматривать распределение (4) при постоянной температуре, что соответствует постоянному спектру накапливаемых нейтронов, так как потери энергии при упругом рассеянии на длине свободного пробега пренебрежимо малы. При таком рассмотрении величина α_g определяет вероятность захвата нейтрона спектра деления в процесс деления, а величина $\nu / (1 + \alpha_g)$ (где ν — число мгновенных нейтронов на акт деления) определяет коэффициент размножения нейтронов на один акт деления.

$$k = \nu / (1 + \alpha_g). \quad (6)$$

На рис. 4 приведены известные экспериментальные данные по величине ν для U^{238} . Так как температура ядра U^{238} при захвате нейтрона с энергией спектра деления колеблется

Рис. 2. Сечение реакции (n, γ) на U^{238} Рис. 3. Величина $\alpha = \frac{\sigma_{(n, \gamma)}}{\sigma_f}$

незначительно (рис. 5), вычислено значение α_g при $T = 1,3$ МэВ, которое для этого спектра равно 90,804, что при максимальном значении $\nu = 4$ (рис. 4) дает коэффициент размножения нейтронов в сборке $k < 0,0435$, и накопление нейтронов в сборке не превышает величины

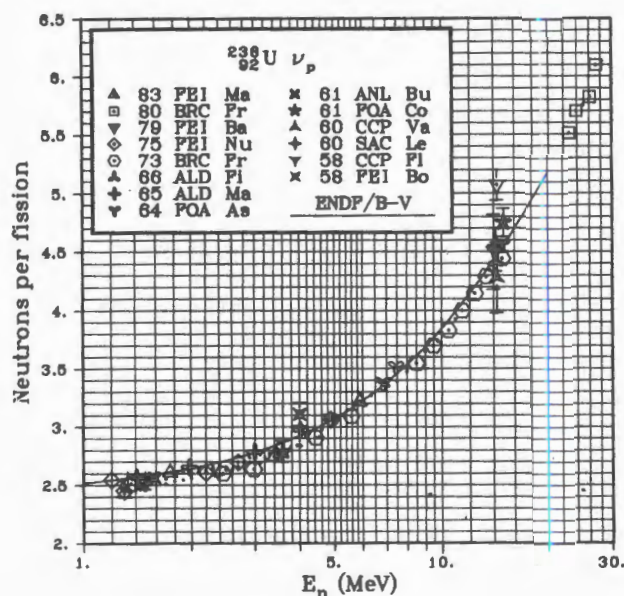


Рис. 4. Количество мгновенных нейтронов на акт деления на U^{238} в зависимости от энергии нейтрона, вызвавшего деление

$$1 + k + k^2 + k^3 + \dots \leq \frac{1}{1 - k} \quad (7)$$

При таком незначительном коэффициенте накопления перспектива использования чистого урана для энергетического усилителя отсутствует. Однако она появляется при увеличении примеси U^{235} с 0,7% до нескольких процентов, при которой величина k в основном определяется этой примесью.

Целесообразность исследования (теоретического и экспериментального) такой возможности диктуется достаточно элементарными соображениями общего плана. Так как тепловая энергия, которая выделяется при одном акте деления, соответствует ~ 190 МэВ, или $3,04 \cdot 10^{-11}$ Дж, для получения тепловой мощности 1 ГВт необходимо «сжигать» около 415 кг U^{238} ежегодно. Если размеры мишени из урана приблизительно соответствуют размерам реактора ($R = 1,5$ м, $\ell = 2$ м), то весовое количество урана будет составлять ~ 260 т, что при сжигании 0,415 т/год будет соответствовать выгоранию 4,8% за 30 лет.

Привлекательность такой схемы очевидна, так как в ней решаются сразу две задачи — подкритичность и существенное уменьшение количества радиоактивных отходов. Однако для разработки схемы потребуется обширный комплекс физических исследований как теоретических, так и экспериментальных, которые сводятся к следующему.

1) Расчет нейтронных потоков в объеме сборки при известном спектре испарительных нейтронов внешнего источника.

2) Определение подкритичности сборки при различном содержании U^{238} , равномерно распределенного по объему сборки.

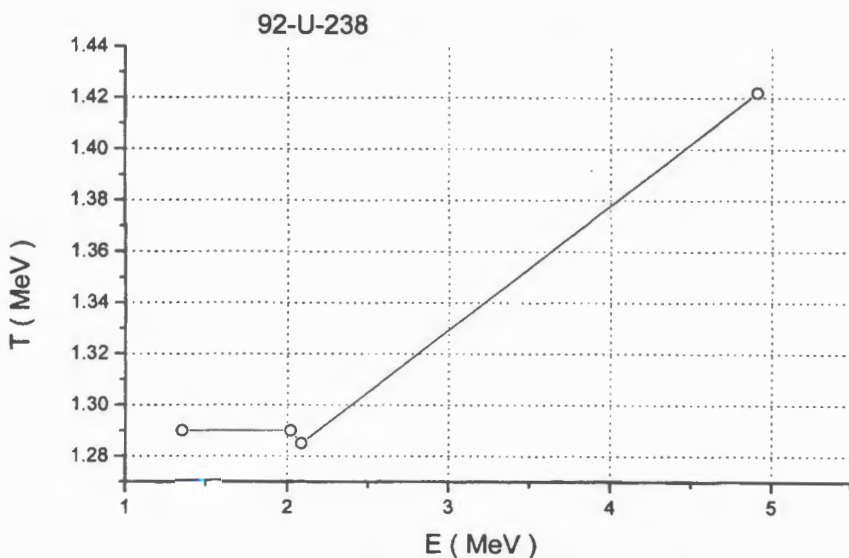


Рис. 5. Температура ядра U^{238} при делении нейтронами различных энергий

- 3) Расчет и экспериментальное определение распределения Pu^{239} по объему сборки, который образуется в реакции n, γ .
- 4) Распределение продуктов деления по объему сборки и их влияние на медленное изменение подкритичности.
- 5) Разработка системы теплосъема, которую можно эксплуатировать в течение 30 лет.

ЗАКЛЮЧЕНИЕ

В работе рассмотрена возможность разработки новых технологических систем для ядерной энергетики, не использующих цепные реакции размножения нейтронов при наличии внешнего источника. Переработка радиоактивных отходов происходит внутри сборки, которая должна увеличить на порядок эксплуатационное время жизни сборки.

В качестве исходного продукта для «сжигания» рассматривается изотоп урана-238, запасы которого в сотни раз превышают запасы ядерного горючего, применяющегося в ядерной энергетике.

Работа выполнена по договору №7/99 с отделом «Дубна» ФИАН им. П.Н.Лебедева.

Литература

1. Rubbia C. — Contribution to the Las-Vegas conf., 1994.
2. Толстов К.Д. — Краткие сообщения ОИЯИ, 1993, № 5(62)-93, с.5.
3. Барашенков В.С., Тонеев В.Д., Чигринов С.Е. — АЭ, 1997, т.37, с.480.

4. Carminati F. et al. — CERN/AT/93-74(ET), 1993.
5. Brandt R. et al. — JINR Comm. E-1-97-349, Dubna, 1997.
6. Брандт Р., Кривоустов М.И., Чултем Д. и др. — Сообщение ОИЯИ Р1-99-117, Дубна, 1999.
7. Bowman et al. — Nucl. Instr. Meth., 1992, A-230, p.336.
8. Дмитриевский В.П. — ЭЧАЯ, 1997, т.28, вып.3, с.815.
9. Дмитриевский В.П., Сергеева Н.В. — Сообщение ОИЯИ Р-9-98-73, Дубна, 1998.
10. Rubbia C., Andriamonje S., et al. — CERN/LHC/98-02(EET), 1998.
11. Субботин В.И. — ЭЧАЯ, 1998, т.29, вып.2, с.333 (136).
12. Михайлов В.Н. — Краткие сообщения ОИЯИ, 1997, № 6, с.17.