ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Жәңгір хан атындағы Батыс Қазақстан аграрлық-техникалық университеті

ЖОО-ның техникалық мамандық студенттеріне арналған физикалық процестерді компьютерлік модельдеу

МОНОГРАФИЯ

Орал 2013

УДК 53: 004.9
ББК : 32.81
Б 65

Жәңгір хан атындағы Батыс Қазақстан аграрлық - техникалық университетінің Ғылыми – техникалық кеңесі ұсынған №4 хаттама, 2013 жыл 26 желтоқсан

Пікір жазғандар:

Доцент.,физ-мат.ғ.к. Кусаинов Р. К
доцент., п.ғ.д . Бахишева С.М.

А. М. Бисенгалиева
И. М Бапиев
А.Х. Касымова

ЖОО-ның техникалық мамандық студенттеріне арналған физикалық процестерді компьютерлік модельдеу . Монография.-Орал: Жәңгір хан атындағы
БҚАТУ, 2013.- 128 б.

Б65 Монографияда модельдеудің компьютерлік бағдарламасы студенттерге физикалық құбылыстарды өз бетімен зерттеуге және сонымен бірге қажетті практикалық дағдыларды қалыптастыруға мүмкіндік береді.

Монография магистранттар, студенттер, кәсіптік білім алушылар, колледж білім алушыларына арналған.

МАЗМҰНЫ

Кіріспе

Компьютерлік модельдеу, есептеу тәжірибесі – басқа зерттеу әдістерімен салыстырғанда өзіне тән ерекшеліктері, артықшылықтары және кемістіктері бар күрделі объектілер мен жүйелерді оқып үйренудің заманауи әдістері. Сәйкесті бағдарламалармен жабдықталған дербес компьютер бірнеше секунд ішінде дифференциальдық теңдеулер жүйесін шешуге, зерттелетін тәуелділік графигін тұрғызуға, зерттелетін үрдісті модельдеуге мүмкіндік береді. Өзінен өзі түсінікті жоғары оқу орындарының студенттері компьютерлік модельдер, танымның әр түрлі объектілерін зерттеудің сандық әдістері туралы хабары болуы керек, қазіргі заманғы бағдарламалық өнімдері білуі қажет. Оқу жүйесінің жаңа түрі – кредиттік жүйе өз бетімен оқу деңгейін жоғарылатуға бағытталған. Сонымен қатар, оқытушымен жұмыс жасауға берілетін уақыт қысқартылады. Көптеген физикалық тәжірибелер оқу зертханаларында өткізуге және зерттеуге мүмкіндік бермейді. Ол бірінші кезекте тәжірибенің күрделілігімен және өте күрделі зертханалық құрал жабдықтарды қажет етумен байланысты.

Модельдеудің компьютерлік бағдарламасы студенттерге физикалық механикалық құбылыстарды өз бетімен зерттеуге және сонымен бірге қажетті практикалық дағдыларды қалыптастыруға мүмкіндік береуі өзектіболып табылады. Оқу үрдісінде компьютерлік технологияны пайдаланудың негіздерін қарастыруды, әр түрлі физикалық құбылыстардың компьютерлік модельдерін жасау әдістері дербес компьютерлерді нақты және есептеу тәжірибелерінде пайдалану әдістемелерін қарастыру.

Негізгі міндеттер:

1. Білім берудегі информациялық технологияларды шолу.
2. Физикалық құбылыстарды математикалық және компьютерлік модельдеу әдістеріне шолу.
3. Нақты оқу есебін қою, шешу және математикалық моделін жасау.

Оқу есебін Delphi-7 ортасында модельдеу.

Монографияда теориялық, практикалық және методикалық негізі механиканың, электродинамиканың теориялық негіздері, математикалық анализ, дифференциалдық теңдеулер теориясы, Delphi-7 бағдарламалық типі бойынша программалап практикалық негізін есептеп шығару болып табылады.

Математикалық модельдеу, аналитикалық және дифференциальдық геометрия, теориялық механика, есептеу тәжірибелерінің әдістемелері пайдаланылды. Техникалық мамандықтар студенттерін физика-механикалық процестерді модельдеу негізінде сәйкесті пәндерді оқуға мүмкіндік беретін виртуаль зертханалық жұмыстардың комплекстерін

құрастырып қолдану. Физикалық құбылыстарды компьютерлік модельдеу және оқытудың заманауи ақпараттың технологияларын қолданудың әдістемелерін талдау. Ойлау іс-қимылдарын біртіндеп қалыптастыру теориясын қолдану негізінде компьютерлік зертханалық жұмыстарды орындау кезінде теориялық модельдерді оқу әдістемесін жасалуы.

Физиканың теориялық модельдерін окып-үйренудің дағдыларын қалыптастыру үшін дидактикалық құралдардың бағдарламалық комплексін жасау, мамандарды дайындаудың сапасын арттыру мақсатында оқу барысында механикалық процесстерді компьютерлік модельдеу пайдалану.

Тарау 1. 1. Теориялық бөлім

1.1. Білім берудегі ақпараттық жүйелер

Білім беруде ақпараттық технологияларды енгізудің негізгі бағыттары: оқытудың үрдісін жетілдіру, оның сапасы мен тиімділігін арттырудың құралы ретінде пайдалану: компьютерлік техниканы білім алудың, өзін танудың және ортаны танудың құралы ретінде пайдалану; компьютерлік техниканы және ақпараттық технологияның жаңа құралдар оқып-білудің объектісі; жаңа ақпараттық технологияны білім алушының шығармашылық дамуының құралы ретінде қолдану; компьютерлік техниканы бақылау үрдісін, түзету, тестілеу және психодиагностика құралы ретінде, педагогикалық тәжірибемен әдістемелік және оқу әдебиетімен алмасу құралы ретінде; ақпараттық техниканы қолдану арқылы коммуникациялар ұйымдастыру; қазіргі заманғы ақпараттық технологиялар құралдары интеллектуальды бос уақытты ұйымдастыру; оқу орнын және оқу үрдісін қазіргі коммуникациялық технологиялары негізінде интенсификациялау және жетілдіру [1].

Қазіргі физикалық зерттеу зертханалары компьютерлермен мейлінше жабдықталған. Қазіргі кезде ғалымдар компьютерсіз қалай жұмыс істейтінін көз алдына елестету қиын. Қазіргі заманғы құралдармен алынған ақпарат көлемі өте көп болғандықтан, оларды алдын ала компьютермен автоматты өңдемей игеру іс жүзінде мүмкін емес. Компьютерлер өлшеу нәтижелерін өңдеу кезінде де , сол сияқты тәжірибені жасау кезінде де баға жетпес көмек көрсетеді. Тәжірибелік қондырғыларды және тәжірибе барысын компьютерлік басқару күрделі қондырғылармен жабдықталған зертханаларды Интернет пайдалану арқылы жетілдіру.

Компьютерлерге байланысты тек ғылыми емес, оқу физикалық зертханалары өз келбеттерін баяу болса да, өзгеретін келеді. Тәжірибелерді компьютерлік басқару, оқу зертханаларында да тәжірибе нәтижелерін автоматты түрде компьютерлік өңдеу қажеттілігі туындап отыр, жақын аралықта бұл бағыттағы өзгерістер міндетті түрде болады. [2,3]

Білім алушыларға инженерлік бағытта оқыту барысында зертханалық практикум міндетті құрамды бөлігі болып табылады. Практилық тұрғыды студенттер теориялық білімдерін практикалық жұмыстарда бекітеді, бақылау және өлшеу құралдарымен жұмыс істеп үйренеді, ізденімпаздық дағдылары қалыптасады.Ақпараттық технологиялармен байланысты, "виртуальды зертханалық практикум" (ВЗП) [4] ұйымы қалыптасты, оның негізін иммитациялық компьютерлік модельдеу алынады. Оқу үрдісінде ВЗП қолдану бағыттары:

- Нақты зертханада практикумды орындауға әзірлік ретінде компьютерлік "тренажер" ретінде қолдану, бұл жағдайда компьютерлік бағдарлама мен физикалық тәжірибенің барысы, көп жағдайда бірдей болады.
- Физикалық қондырғыда әр түрлі себептермен (техникалық, қаржылық, ұйымдастыру және т.б.) жүзеге асыру мүмкін болмағанда, нақты практикумға қосымша ретінде компьютерлік тәжірибелермен толықтыру.

Компьютерлік "тренажер" ретінде ВЗП қолдану физикалық тәжірибеге білім алушының жан-жақты дайындалуына [5], зерттелетін құбылыстарды терең игеруіне, өлшеуіш құралдармен жұмыс істеу дағдыларының қалыптасуына (бұл жағдайда виртуаль практикумды нақты приборларға қасиеттері жақын өлшеуіш құралдар пайдаланған жағдайда). Мұндай жағдайда көбіне сырттай – дистанциялық оқыту бөлімінің студенттері ұсынылады, себебі оқылатын материалды терең игеруге мүмкіндік беріп қоймайды, сонымен қатар оқу орнында нақты зертханаларда практикумды орындау уақытын қысқартуға оң әсерін тигізеді.

Егер ВЗП нақты практикумға толықтыру ретінде қолданылса, ол күрделігі жоғары практикумды орындауға немесе университетте күрделі құрал-жабдықтар болмаған жағдайда зерттеу жұмыстарын жүргізуге бағытталуы керек.

ВЗП жабдықтау технологиясы бойынша негізгі нұсқаларды бөліп алуға болады [6].

1. Бірнеше пәндердің үлкен аумағында қолдануға болатын бағдарламалардың әмбебап қажеттері негізінде ВЗП. Мысалы: National Instruments фирмасының LabVIEW жүйесі [4,6] (ағылш. Laboratory Virtual Instrumentation Engineering Workbensch). Әмбебап пакеттерге физикалық құралдар мен зертханалық қондырғылардың виртуаль интерфейстерін жасауға қажетті элементтердің аумақты кітапханасы болады.
2. Пән – бағытталған арнайы бағдарламалар пакеттердің негізінде біршама шектелген пәндер жиынына арналған ВЗП. Мысалы: Elektronics Workbensch фирмасының электрондық сұлбаларды модельдеу үшін жасалған Multisim жүйесін, химиялық үрдістерді

талдау және модельдеу үшін Gambridge Soft фирмасының Ghem office жүйесі және т.б. алуға болады. Бұл топтың бағдарламалық қамтуы, алдыңғы топтағы сияқты, пайдаланушының қолданбалы есептерін шығаруға арналған әмбебеп орта болып табылады.

3. Java – апплеттер негізінде ВЗП. Оқытушы графикалық бағдарламалар жүйесінде істейтін алдыңғы қарастырылған жағдайлардан өзгеше, Java – апплет жасау үрдісі көп жұмыс жасауды және бағдарламалық кодты жасауды талап етеді. Осыған қарамай, бұл технология желілік қолдануға арналған ВЗП туралы сөз болғанда артықшылықтары болады. Мысалы: LabVIEW жүйесінде қосмымшалар жасағанда 2,5-3 Мбайт көлемінде жады қолданылады, Java – апплет негізіндегі зертханалық жұмыстардың көлемі ондаған – жүздеген килобайт болады.

Желілік компьютерлік технологиялық прогрессивті жетілуі нақты құрал-жабдықтан қашықтан оқыту режимінде жүзеге асырылатын зертханалық практикумның жасалуына мүмкіндік береді. Нақты құрал жабдықтарға қашықтан әсер етуді жүзеге асыру бірнеше мәселелерді шешуімен байланысты болғандықтан (зертханалық макетті ДК – үйлестіру, қондырғыларды апаттық жағдайлардан сенімді қорғау, қондырғыларға ұжымдық әсер етудің кей жағдайда мүмкін болмауы және т.б.), бұл технологияға қарсылық білдірушілер де бар. Дегенмен, бұл жүйенің ВЗП қарағанда кейбір артықшылықтарының болуы, оны да пайдаланудың мүмкіндігі бар екендігін көрсетеді.

ВЗП нақты лабораториялық практикумға альтернативасы болмай ма деген сұрақ та маңызды [7,8]. Бір жағынан, қазіргі имитациялық модельдеудің компьютерлік технологиялары нақты зертханалық қондырғылардың сыртқы пішінін, оның параметрлерін жоғары дәлдікпен жасайтын виртуаль интерфейстер жасауға мүмкіндік береді. Екінші жағынан, қазіргі зертханалық қондырғыларды және өлшеуіш құралдарды жұмыс жағдайында ұстап отыру және өз уақытында жаңартып отыру үшін біршама қаржы қажет етеді.

Қазіргі кездегі концепциялардың біреуін сай, адам қабылдайтын ақпарат төмендегі сатылардан өтеді: сенсорлы-моторлы, символды, логикалық және лингвистикалық.

Бірінші сатыда ақпаратты түйсікті қабылдау болады, екінші этапта оны образдарға түрлендіру, үшіншіде – оны түсіну, төртіншіде – ақпарат санада "сөз-образ" арқылы тіркеледі.

Табиғаттану ғылымдары пәндерін оқытуда әр уақытта да сенсорлы-моторлы саты орын алады. Мұнда зертханалық практикумдар және оқу тәжірибелері елеулі роль атқарады [9].

Оқу тәжірибесі білім беру әдістерінің маңыздыларының бірі болып табылады. Ол қандай да болмасын курстық кіріспесі ретінде қолданылуы мүмкін (мотивация), жаңа материалды түсіндіруге иллюстрация (қабылдау

және түсіну), өткен материалды қайталау және қорытындалау (интериоризациялау), немесе оқытуды барлық кезеңінде алған білімді, дағдылады т.б. бақылау ретінде пайдаланылады.

Оқу тәжірибелері

Демонстрациялық тәжірибе**:** Заңдарды, құбылыстарды түрлендіру кезінде иллюстрациялау және оларды қодануы көрсету, техникалық қондырғыларды жұмыс істеу принципін, іргелі тәжірибелермен оқушыларды таныстыру үшін қолданылады. Оны тек оқытушы әзірлеп, өткізеді.

Фронталь зертханалық жұмыстар, тәжірибелер және бақылаулар. Сабақ барысында оқытушы басқаруымен барлық студент бір мезгілде бірдей қондырғылармен жасайды [10].

Практикалық тұрғыдан білім алушылардың өзбетімен жұмыс істеу формасы болып табылады, білім алушылар алдын-ала әзірлеп, жазбаша нұсқаулықтар бойынша орындайды.

Сабақтан тыс тәжірибелер және бақылаулар. Үй тапсырмасының бір түрі. Теориялық білімдерді өзектілеу және жаңа материалды оқуға ынталандыру мақсатында пайдаланылады.

Әдістемелік жағынан дұрыс ұйымдастырылған тәжірибе практикалық білім қалыптастыру үшін де, бұрын алған теориялық білімдерін белсендіре түсуге де көмек көрсетеді. Оқыту процесінде қабылдаудың әр түрлі арналары (есту, көру, сезіну т.б.) іске қосылады. Бұл алынған ақпаратты жарқын бейнелер жүйесі түрінде қалыптастырып, ұзақ еске сақтауға мүмкіндік жасайды. Екінші жағынан, зертханалық жұмыстарды дайындау және орындау оңай іс емес, оқытушыдан белгілі бір әдістемелік ерекшеліктерді білуді талап етеді, белгілі бір приборлар мен аспаптардың болуын қажет етеді.

Осы және басқа да мәселелерді белгілі бір деңгейде білім беру кезінде компьютерлік тәжірибелер арқылы шешуге болады. Бұл термин зертханалық жұмысты басқа да техникалық құралдарды пайдаланбай компьютерде толықтай атқаруды қамтиды.

Оқу үрдісінде компьютерлер қолдану арқылы тәжірибе жасау көптеген ұсақ мәселелерді шешуге (сұлбаның параметрлерін өзгерту, тәжірибе нәтижесін өлшеу, есептеу т.б.) қажетті уақытты елеулі қысқартуға мүмкіндік береді, сөйтіп өткізілген тәжірибенің мақсаты мен міндеттеріне елеулі көңіл бөлуге мұрса береді. Сонымен қатар оқу кабинетінде жасауға келмейтін тәжірибе көрсету мүмкіндігі пайда болады [9, 10].

Компьютерлік тәжірибеге тән ерекшеліктерін көрсетейік. Формасы – оқушы мен компьютердің функциялары төмендегідей:

- қарастырылатын объектінің, қондырғының, процесстің немесе күйдің моделін бағдарламалардың көмегімен жүзеге асыру;

- өлшеу құралдарын имитациялау және өлшеу нәтижелерінің ұсақ-түйек бөлігін орындау;
- оқушылардың іс-қимылдарын бағалау;
- оқушының міндеттері (дәстүрлі тәжірибедегі оқушыныің міндеттерінен өзгеше);
- бағдарлама дисплей экранына шығарған ақпаратты талдау;
- эксперименттің шарттарын таңдау;
- жұмыстың басында тұжырымдалған мақсаттарға жету үшін тәжірибелер сериясын өткізу;
- жоғары баға алу мақсатында және есепті тиімдірек жолмен шығару келесі қадамдарға түзетулер енгізу.

Экспериментті тек компьютерлік нұсқауда жүзеге асыру мүмкіндігі анық. Бірақ та, компьютерлік тәжірибенің дидактикалық тиімділігі және қызықтылығымен қатар кейбір мәселелер шешілмей қалады. [10, 11].

Біріншіден, ақпаратты қабылдау дәстүрлі зертханалық жұмысты қабылдаудан елеулі өзгеше болады, жеке жағдайда сенсорлы-моторлы саты болмайды. Бұл сатысыз қабылдау толық бола алмайды. Нәтижесінде оқыту да толық болмайды.

Екіншіден, нақты құралдар мен қондырғылар жұмыс жасау кезінде алатын политехникалық дағдылар жөнінде мәселе туындайды.

Нақты емес объектілермен жұмыс істеу кезінде ерекшіліктер туралы сәйкесті көзқарастарды қалыптастыру мәселесі өте маңызды және әлі толық зерттелмеген.

Бұл мәселені шешу нақты дүниедегі үдерістер мен құбылыстарды барынша толық бейнелейтін бағдарламаларды оқу үрдісінде пайдалану арқылы шешуге болатын болар. Бұл жағында виртуаль шындықтың құралдарына аса көңіл аудару керек. Сонымен табиғаттану ғылымдары циклін оқытудың негізгі мәселесі оқу және зертханалық тәжірибелерді нақты түрде қоюдың шектелуі болады.

Қазіргі жағдайда виртуаль-зертханалық практикалық жұмыстары болғанмен, мәселені түпкілікті шешу үшін әр түрлі салада мамандарды, психолог-педагогтардың баса назар аударғаны жөн.

Жоғары оқу орындарының зертханалық және тәжірибелік базалары өзгеріске баяу түседі, материалдық және қаржылық шектеулер оны тез өзгертуге тежеу болады. Сондықтан олар техниканың қарқынды дамуына ілесе алмайды, құрал-саймандар моральдық жағынан тез ескіреді. Қазіргі тез өзгеретін жағдайда мамандықтар мен мамандандарылу өндірістің сұранысына жылдам және үздіксіз бейімделуі керек, жоғары оқу орнының зертханалық және тәжірибелік базасы оқу үрдісін қажетті деңгейде ұстап тұра алмайды [12]

Виртуаль приборлардың қазіргі технологиясы осы артта қалуды азайтуға және оқу сапасын төмендетпей елеулі қаржылық ресурстарды үнемдеуге мүмкіндік береді. Мысалы, виртуаль приборлар технологиясын

қолдайтын және соған сәйкесті оқу зертханаларын тез икемделетін бағдарламамен қайта құрылатын өлшеуіш құралдармен қамтамасыз етеді немесе зертханада кез-келген күрделі өлшеу құралдарын модернизациялау, жоғары оқу орнына автоматтанған өлшеуіш жүйелер мен станцияларды оқу үрдісіне енгізуге мүмкіндік береді.

LabVIEW бағдарламасы [4,13] да виртуаль құрал болып табылады (ағылш. Virtual Instrument), екі бөліктен тұрады:

- виртуаль құралдың жұмыс логикасын сипаттайтын блокты диаграмма;
- виртуаль құралдың сыртқы интерфейсін сипаттайтын алдыңғы панель.

Виртуаль құралдарды басқа да виртуаль приборлар жасау үшін құрамды бөлігі ретінде қолдануға болады.

Виртуаль құралдың алдыңғы беті кіріс-шығыс құралдарынан тұрады: түймелер, ауыстырып-қосқыштар, жарық диодтары, верньерлер, шкалалар, ақпараттық табло және т.б. оларды білім алушылар виртуаль құралды басқару үшін және басқа виртуаль құралдармен ақпарат алмасу үшін қолданады.

Блокты диаграмма берілген шамадарды өңдеу құралдары, көзі, қабылдағыш болатын функционал тораптардан тұрады. Блокты диаграмманың компоненті ретінде терминалдар алдыңғы панельдің объектілерінің "артқы контактылары", басқару тетіктері "IF шартты операторы, FOR және WHILE т.б." түрінде бағдарламаларды текст элементтерінің аналогтары.

LabVIEW көптеген өндірушілердің құралдарының зор спектрін қолдайды, компоненттердің көптеген кітапханасы болады:

- Көп тараған интерфейстерге және протоколдарға (RS – 322, GPIB – 488, TCP/IR және т.б.) сыртқы құралдарды қосу үшін;
- Тәжірибенің жүрісін алыстан басқару үшін;
- Роботтар мен машина жүру үшін жүйелерін басқару үшін;
- Сигналдар шығару және цифрлық өңдеу үшін;
- Мәліметтерді өңдеудің әр түрлі математикалық әдістерін қолдану үшін;
- Нәтижелерді көру және оларды өңдеу (3Д модемдерді қоса);
- Күрделі жүйелерді модельдеу үшін;
- Мәліметтер барысында ақпаратты сақтау және есептерді генерациялау;
- COM/DCOM/OLE концепциясының аумағында басқа қосымшалармен өзара әсерлесу үшін.

LabVIEW арнайы бөлігін Application Buildez басқа компьютерлерде қолдануға болатын LabVIEW бағдарлама жасауға мүмкіндік береді. Бұл бағдарламалар жұмыс жасау үшін тегін таратылатын "LabVIEW Runtime

Engine", болады, қажет болған жағдайда қолданылатын сыртқы қондырғылар драйвер керек.

20 жылдан астам инженерлер мен ғылымдар NI LabVIEW өлшеуіш жүйелерді, сынақ стенділердің және басқару жүйелерін жасау үшін қолданып келеді. LabVIEW негізінде G бағдарламасының графиктік тілі жатады. LabVIEW ортасы бағдарламалау мүмкіндігінен басқа тұтынушыға құралдар мен кітапхананың кең спектрін береді: реттеуді интерактив шеберінен, пайдаланушы интерфейсінен бастап жете орналасқан компилятор, құрастырушы және реттеу құралдарына дейін.

Оқу процесіне виртуаль құралдар мен өлшеу жүйелерін технологиясын іске асыру үшін кіріс-шығыс аналогты стандартты платасын алу жеткілікті, оның негізгі құрамды бөлігі көп арналы коммутатор және аналогты-цифрлы түрлендіргіш.

Сандық эксперимент – физикалық құбылыстың математикалық моделін жасап, осы модельді сандық зерттеп, оның әр түрлі жағдайдағы өтуін модельдеу [14].

Компьютерлік модельдеу жол іздеген ғалымға да, болашақ мамандарға жаңа білім алуға талпынған білім алушыларға да елеулі көмек көрсете алады. Оқу мақсатындағы модельдеуден компьютерлік бағдарламалар дәстүрлі қолданысты электронды түрде қосымша ғана емес, зерттелетін физикалық құбылыстың математикалық моделін пайдаланатын және интерактивті жұмыс істеуге арналған шағын зертхана болып келеді. Бұл жағынан алғанда модельдеуші бағдарламалардың дәстүрлі оқу-бақылау компьютерлік бағдарламалардан ерекшелігі, атап айтқанда, физикалық құбылыстарды модель дегенде компьютерлердің кең мүмкіндігі пайдаланылады. Модельдеуші бағдарламалармен студенттің қатысуымен өтетін кішігірім ғылыми зерттеуге ұқсайды [2,15].

1.2. Модельдеу теориясы. Негізгі қағидалар

Көп мөлшердегі физикалық тапсырмалар математикалық физика теңдеулері түрінде формулданады, себебі көбіне аналитикалық түрде шешім алу сәтті болмайды.

Модель (Model, simulator) –

1) қасиеттері белгілі бір мағынадағы жүйенің немесе процестің қасиеттеріне ұқсас объектілер немесе процестер жүйесі;

2) сериялы бұйымдарды жаппай өндіруге арналған үлгілері мен эталон; кез- келген бір объект жұмысы, мысалы, процессордың жұмыс істеуін модельдейтін программа немесе құрылғы. Ол материалдық объект түрінде, математикалық байланыстар жүйесі ретінде немесе құрылымды имитациялайтын программа түрінде құрастырылады да, қарастырылатын объектінің жұмыс істеуін зерттеу үшін қолданылады. Модельге қойылатын негізгі талап – оның

қасиеттерінің негізгі объектіге сәйкес келуі, яғни парапарлығы. Модельдеу (Simulation) – кез-келген құбылыстардың, процестердің немесе объект жүйелерінің қасиеттері мен сипаттамаларын зерттеу үшін олардың үлгісін құру (жасау) және талдау; бар немесе жаңадан құрастырылған объектілердің сипатын анықтау немесе айқындау үшін олардың аналогтарында (моделінде) объектілердің әртүрлі табиғатын зерттеу әдісі. Модель төрт деңгейде түпнұсқаның гносеологиялық орынбасары бола алады: 1-элементтер деңгейінде, 2-құрылым деңгейінде, 3- қалып-күй немесе қызметтік деңгейінде, 4-нәтижелер деңгейінде. Сипаты бойынша модельдеу материалдық және идеалдық болып бөлінеді.

Материалдық модельдеу объектінің геометриялық, физикалық, динамикалық және қызметтік сипатын нақты дәл береді.

Модель дегеніміз - нақты объектіні, процессті немесе құбылысты ықшам әрі шағын түрде бейнелеп көрсету.

Модельдеу – объектілерді, процесстерді немесе құбылыстарды зерттеу мақсатында олардың моделін (макетін) құру.

Кез келген жұмысты қолға алмас бұрын, берілгені мен соңғы нәтиже және орындалатын іс-әрекет кезеңдерін айқындап алу қажет. Модельдеу кезінде бастапқы зерттелетін объект – прототип болады. Модельдеудің соңғы кезеңі шешім қабылдау болып табылады. Модельдеу арқылы зерттелген модельдің жаңа объектісін құруға, бар объектіні жақсартуға немесе қосымша ақпарат алуға болады. Модельдеудің негізгі кезеңдері есептің қойылу шарты мен мақсатына қарай анықталады.

1-кезең Есептің қойылымы. Бұл кезеңде берілген бастапқы мәліметтермен қатар мақсатын анықтау және объектіні немесе процесті талдау анық көрсетілуі қажет.

2-кезең Модель құру. Ақпараттық модель. Бұл кезеңде элементар объектілердің қасиеттері, күйі және басқа да ерекшеліктері кез келген пішінде, яғни ауызша түрде, схема немесе кесте арқылы да анықталады. Бастапқы объектіні құрайтын элементар объектілер жөнінде толық мағлұмат, яғни ақпараттық модель жасалады. Бұл кезең модель құрудың бастапқы бөлімі болып саналады.

3-кезең Компьютерлік эксперимент. Жаңа конструкторлық жұмыс, техникалық шешімдерді өндірісте пайдалану және жаңа идеяларды тексеру үшін эксперимент жасау қажет. Компьютерлік тәжірбие жүргізу екі кезеңнен тұрады: модельдеу жоспарын құру және модельдеу технологиясы. Модельдеу жоспары модельмен жасалатын жұмыстың ретін анық көрсетуі қажет. Модельдеу технологиясы дегеніміз – пайдаланушы адамның компьютерлік модельмен орындайтын мақсатты іс-әрекеттерінің жинағы.

4-кезең Модельдеу нәтижесін талдау. Модельдеудің соңғы мақсаты – шешім қабылдау болып табылады. Модельдеу нәтижесін

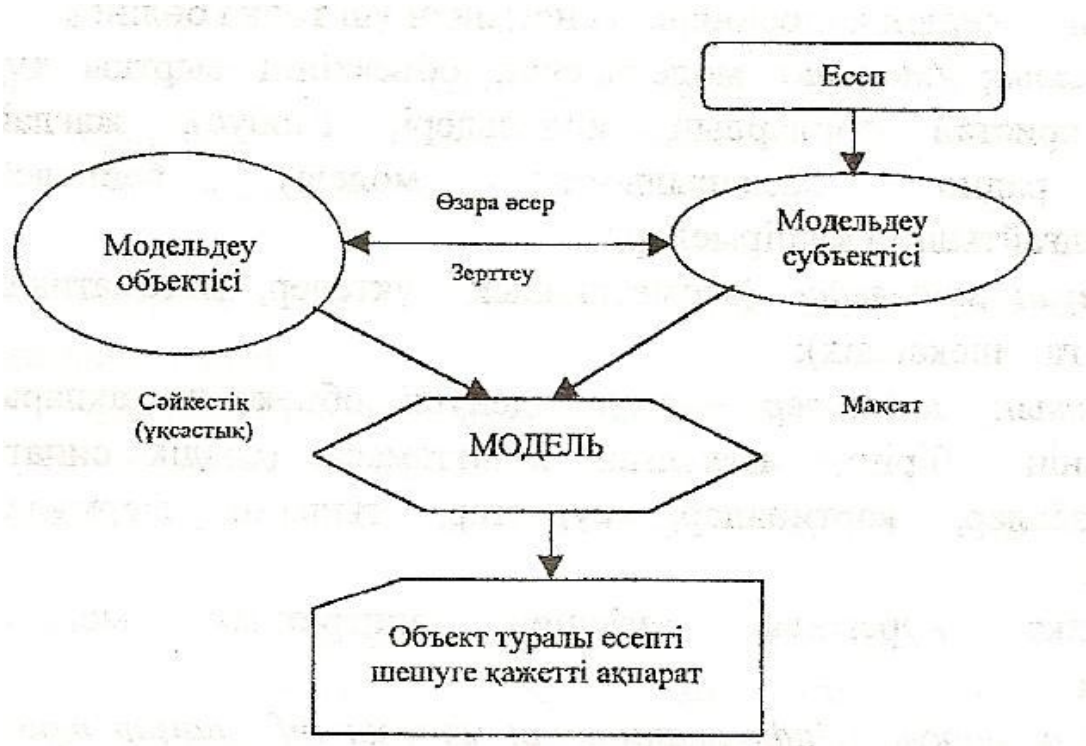
талдау шешуші кезең болып табылады. Себебі, бұдан кейін модельдеуді жалғастыру немесе тоқтату керек. Компъютерлік модельдеу - бұл да оқып үйренетін объекті теориясының модельденуі.

Моделъдеуші (модель субъектісі) тек адам бола алады. Модельдеу объектісі табиғи (өсімдік, күн жүйесі) және адамның ықпалымен құрылып жасанды болуы мүмкін (1.1-сурет).

Модельдеу жүйесі (todeling systet) - зерттелетін жүйенің немесе оның элементтерінің математикалық және· физикалық аналогтарын құру және талдау. Модельдік тәжірибе зерттеу тәсілі ретінде жүйені жаңғыртуға және зерттеуге мүмкіндік береді, ал зерттелетін жүйеге тікелей тәжірибе жүргізу қиын, немесе экономикалық тұрғыдан тиімсіз болуы мүмкін [l3].

Табиғи объектілерді ешқандай моделъдің толықтай бейнелей алмайтындығы белгілі. Табиғи объектілердің элементтерінің арасындағы байланыстардың көбінесе белгісіз болуы олардың күрделілігін айқындайды. Сондықтай табиғи объектілердің модельдері түп нұсқаға қарағанда қарапайым болады. Адамдар тарапынан құрылатын объектілерде мұндай жағдайлардың толық ескерілмеуі мүмкін.

1.1-сурет. Модельдеудің жалпы схемасы.



Бірақ модельдеу барысында моделъдеу мақсаты тұрғысынан қажетсіз детальдар еленбейді.

Модельдер не үшін қажет?

Адамның практикалық ғылыми қызметтерінде жұмыс істеуіне тура келетін объектілердің қандай да бір алмастырушысын құрады.

Мұның табиғи көшірме -картина скульптура; самолеттің ұшу қасиетін зерттеуге бел-гіленген макеті; қандай да бір бұйымның партиясын дайындауға арналған үлгісі болуы мумкін [l4].

Адамның оқып үйренетін объект туралы ақпараттық моделінің негізін құрайтын қажетті ақпараттарды жинақтауы қажет.

Практикалық есепті шешу тұрғысынан модельдерді пайдалану оқып үйренетін объектілердегі модельдеудің мәнін, мазмұнын демонстрациялауға мүмкіндік береді. "Модель" термині көп мағыналы. Модель деп к:андай да бір заттың кішірейтілген көшірмесі (самолет моделі тұрғын үйлер макеті), математикалық формулаларды, бұрыштан көкжиекке лақтырылған дененің ұшу моделін, іштен жану двигателі жұмысының моделін, бұйымдарды жинау моделін, құрамы бойынша сөйлем талдау моделін, қандай да бір нәрсенің эталонын (метр эталоны, килограмм эталоны) айтамыз.

Жалпы түрдегі "модель" түсінігі төмендегідей негізде анықталады.

Модель – модельдеу мақсаты тұрғысынан оқып үйренетін объектінің, құбылыстың кейбір жақтарын ұқсастырып бейнелейтін жаңа объект.

Модель - объектінің нақты жұмыс істеуіне сәйкестенетін анықталған параметрлер бойынша жұмыс істейтін физикалық ақпараттың алмастырушысы.

Модельдеудегі ең бастысы модельдеуші объекті мен оның моделі арасындағы өзара ұқсас қатысы болып табылады.

Модель – бұл заттық немесе ойдағыны ұсынатын объекті, яғни зерттеу барысында түпнұсқа объектісінің орнын басатын, ол туралы жаңа мәліметтер беретін.

Модельдеу барысында моделдеуді құру процесі, зерттеулері және қол-данулары қарастырылады. Ол абстракция сияқты, ұқсастық, болжам және тағы басқа категориялармен тығыз байланысты. Модельдеу процесі міндетті түрде абстракцияларды құруды, ұқсастықпен ойша пайымдауды, ғылыми болжамдарды құрастыруды қосады.

Модельдеудің негізгі ерекшелігі, ол объекті-орынбасарлары көмегімен ортақталған тану әдісі. Модель танудың өзіне тән аспабы секілді әрекет жасайды, яғни зерттеуші өзін қызықтыратын объектісін оның көмегімен зерттеп біледі. Нақ осы модельдің әдіс ерекшелігі қолданудың ерекше түрлерін, ұқсастықтарды, болжамдарды, басқа категорияларды және тану әдістерін анықтайды.

Модельдеудің екі әртүрлі жолы бар. Модель объекті көшірмесіне ұқсас болуы мүмкін, басқа материалдан, басқа масштабта жасалған, бөлшектік қатарлары жоқ болуымен мүмкін. Мысалы, бұл кішкентай

ойыншық кеме, кубиктардан құрылған кішкене үй, авиаконструрктрда қолданылатын натуральды көлемдегі ағаш ұшақтың үлгісі және т.б.. Мұндай модельдер табиғи деп атайды.

Физикалық модельдеу мүмкіншіліктеріне шек қойылған. Ол тапсырма кезінде үйлестірудің кішкене санын жүйе параметрлерін зерттеуде басқа мақсаттарда шешуге рұқсат етеді.

Яғни, есептеуіш желіде табиғи модельдеудің жұмысын әртүрлі түрдегі коммуникациялы құрылғы –маршрутизаторларды, коммутаторларды және т.б. қолданып тәжірбиеде тексеру мүмкін емес. Тәжірибелер кезінде тексеру ондаған әртүрлі маршрутизатор үлгілері тек үлкен жігерлермен және уақытша шығындармен ғана емес, сонымен қатар - заттық шығындарменде байланысты.

Математикалық модель-математика тілінде берілген модель. Басқаша айтқанда, бұл математикалық объектілердің жүйесі (сандардың, өзгергіштердің, жиындардың, графиктердің, матрицалардың және т.б.) және зерттелетін процестің кейбір қасиеттерін бейнелейтін олардың аралық қатынысы.

Аналитикалық және еліктеу моделдерін айырып тануға болады. Аналитикалықтар болып, алгебралық, дифференциалды және басқа теңдеулерді қолданатын, сонымен қатар олардың дәл шешімін табатын бір мағыналы есептеуіш процедураларды жүзеге асыруды ескеретін нақты объекті модельдері есептеледі. Еліктеулер болып, үлкен сан жүйелі элементарлық операциялардың орындалу жолын зерттейтін жүйенің жұмыс жасау алгоритмі математикалық модель деп аталады.

Модельдеу принциптары келесілер арқылы түзеледі:

1. Ақпараттық жеткіліктілік принципы. Толық ақпарат болмаса обьекті туралы модель құру мүмкін емес. Толық ақпарат алу барысында модельдеу мағынасынан айрылады. Ақпараттық жеткіліктіліктің деңгейі бар, оған жету кезінде модель жүйе құрылылады.
2. Жүзеге асушылық принципы. Жасалынушы модель зерттеудің соңғы уақытына дейін қойылған мақсатқа жетуге тиісті қамсыздандырылуы тиіс.
3. Моделдеудің көптік принципы. Кез-келген нақтылы модель нақты жүйенің тек кейбір жақтарын бейнелейді. Толық зерттеулер үшін зерттейтін процес модельдерінің қатарын құру керек, яғни әрбір келесі модель алдындағыны анықтауға тиісті.
4. Жүйелілік принципы. Зерттейтін жүйе ішкі жүйелердің бір-бірімен бірлесіп әрекет қылатын жиынтық түрінде ұсынылады, яғни стандартты математикалық әдістермен модельденеді. Осыдан жүйе қасиеттері оның элементтерінің қасиеттер қосындысы болмайды.

5. Параметризация принципы. Кейбір модельденуші жүйенің ішкі жүйесі тек бір параметрмен сипатталуы мүмкін: вектормен, матрицамен, графикпен, формуламен.

Есептеуіш тәжірибенің негізгі кезеңдерін сәйкес физикалық тәжірибемен салыстыруға болады. Есептеуіш тәжірибенің бірінші сатыда зерттелетін физикалық обьектіні талдау негізінде оның математикалық суреттелуі жасалады, яғни математикалық моделі таңдалады (физикалық тәжірбиеде бұл сатыға тәжірбиенің схемасын таңдау мен талдау сәйкес келеді, конструция элементін анықтау және өзңндік қоюды). Әрі қарай сандық әдіс таңдалады, есептеуіш алгоритм құрылады, шешімнің дәлдігі және сұрақтардың тұрақтылығы зерттеледі (физикалық тәжірбиеде бұл сатыда тәжірбиелік қойылымды құрастыру және даярлау, оны жөндеу, калибирлеу жүзеге асады) Одан кейін бағдарламалау және машиналық шоттау жүзеге асады (зерттеу шартында бұл сатыға тәжірбиелік өлшемнің сериясын беру талапқа сай болады). Соңғы сатыда талдау нәтижесі және моделді, әдісті, бағдарламаны анықтау жүзеге асады. Мұндай талдау нәтижесі қажетті түзетулермен тәжірбиелік қойылым конструкциялары және тәжірбие схемалары зертханалық шарттарда өткізіледі.

Физикалық, сонымен қатар математикалық модель ді таңдауда негізінен зерттелуші процесске ықпал жасаушы факторларға көңіл бөледі. Физикалық құбылыстарға сәйкес типтік математикалық модельдер математикалық физика теңдеулерімен беріледі. Есептеуіш тәжірибе тәжірбиенің теориясы есептері және зертханалары арасында қызмет етеді. Ол сандық әдістерін дамытуда қолдануда қуатты стимулмен қызмет етеді.

Сандық әдіс арқылы ЭЕМ-да «дискретті модель»-деп аталатын математикалық модель түсіндіріледі. Мысалы, егер математикалық модель дифференциалды теңдеумен өрнектелсе, онда оның әртүрлі теңдеуін аппрокусимиралаушы алгоримтммен бірге, оның шешімін табушы сандық модель болуы мүмкін.

Алгоритмді іске асыру үшін, ЭЕМ–ға бағдарлама құру қажет. Бағдарламаны тексеруден кейін шешімнің есептеулері және талдауларын жүргізу керек. Алынған шешімдер оған сәйкес зерттелетін көрсетіліммен қарастырылады, қажеттілік үшін математикалық модель анықталады, және сандық әдіске дұрыстаулар енгізіледі.

Есептеуіш тәжірбие схемасы жалпы түрде оның негізін триада құрайды: модель- әдіс (алгоритм)-бағдарлама. Ірі есептеуіш тәжірибе қолдануларын көрсетуге болады, яғни энергетика, эрокосмостық техника, ядролық физика.

Айтылған әдістермен суреттелген негізгі обьектіні зерттеу процессінде әр этапта қателіктер жібереді. ММ құру оңайлатылған негізгі обьектімен байланысты, яғни теңдеудің және басқа кірісте берілгендердің

коэффициенттері жеткілікті дәл берілмейді. Бұл қателіктер берілген модель рамкаларында шарасыз және сондықтан сандық әдіске

көңіл бөлумен берілген модельді іске асыруда жойылмайтындай көрінеді. Математикалық модельден сандық әдіске көшуде Әдіс қателіктері атты қателіктер кетеді. Олар кез-келген сандық әдіс математикалық әдісті жуық түрде құруымен байланысты. Әдістің типтік қателіктері болып, Дискретизация қателігі және дөңгелектеу қателігі саналады.

Модель деп қаланың болашақ шағын ауданыныңң жазық графикалық немесе кеңістіктегі көлемді келбетін де айтады. Көптеген адамдар өнер мен әдебиет шығармаларын «адам жанының» моделі дейді [15,18]. Біздің табиғат туралы білімдеріміздің жиынтығы оның моделі деген түсінік те кеңінен тараған. Модельдеу сөзі грамматикалық құрамы бойынша оған кең мағына береді.

Модель дегеніміз бір объектіні (оригинал) екіншісіне (модельге) алмастырып, модельдің қасиеттерін зерттей отырып оригиналдың қасиеттерін зерттеу және бекіту. Объект (жүйе) параметрлер мен сипаттамалардың жиынтығымен анықталады. Жүйенің параметрлерінің жиыны оның мазмұнын – құрылымы мен жұмыс істеу принципін анықтайды.

Жүйенің сипаттамалары – оның басқа жүйелермен өзара әсерлескендегі маңызды сыртқы қасиеттері. Жүйенің сипаттамалары оның параметрлерімен функционалды байланысқан [16].

Модельдеу теориясы модельдерді құрастырып, зерттеу үшін қажет құралдардың, әдістердің, анықтамалардың, қағидалардың өзара байланысқан жиынтығы. Бұл қағидалар, анықтамалар, әдістер, құралдар және модельдердің өздері модельдеу теориясының пәні болып табылады.

Модельдеу теориясының негізгі міндеті – зерттеуші оригиналдың қажетті қасиеттерін жеткілікті дәл және толық анықтайтын зерттеуге қарапайым және жылдам алынған нәтижелердің оригиналға көшіруге мүмкіндік беретін модель жасаудың технологиясын жасау.

Бізді қоршаған физикалық құбылыстар дүниесі өте жан-жақты және күрделі. Осы дүниені тану үшін біз нақты физикалық жүйенің жеңілдетілген математикалық модельдері пайдалануға мәжбүр боламыз [17]. Барлық қасиеттерін ескеру есепті тіпті қарапайым құбылыстар үшін қиындатып жіберер еді. Әрине, таңдалған модель зерттелетін нақты құбылыстың барынша маңызды және тән белгілерін сақтауы қажет. Егер біз физикалық құбылыстың сәйкестігі математикалық моделін жасай алсақ, оны түсіндік деуге болады.

Кейбір жағдайда алынған модельдің шегінде қойылған есептің дәл немесе жуықталған аналитикалық шешуі алуға болады. Өкінішке орай, дәл аналитикалық шешулер физикада сирек кездеседі. Көптеген жағдайда қозғалыстың дифференциалдық теңдеуі интегралданбайтын болып келеді.

Бұл жағдайда көмекке теңдеуді шешудің сандық әдістері келеді және нақты зерттелетін құбылыс туралы көзқарасымыз дұрыстығын компьютерде санау тәжірибесі арқылы тексеруге болады.

Модельдеудің ғылыми зерттеулерде, инженерлік шығармашылықта жалпы адам өміріндегі орнына артық бағалау мүмкін емес. Кез келген жүйені құрастыру және зерттеу шын мәнінде оның моделін жасауға әкеледі. Жүйенің қасиеттерін тіркеумен қатар оның сипаттамаларының жүйенің параметрлеріне тәуелділігін зерттейтін конструктивтік модельдер аса бағалы [18]. Мұндай модельдер жүйелердің жұмыс істеуін оңтайландыруға мүмкіндік береді.

Санау жүйелерін зерттеуге қолдану мүмкіндігін жүзеге асыратын әдістер таңдалады. Материалды талдағанда санау жүйелерінің әр түрлері және олардың құраушы бөліктері алынады.

Объектінің моделі – зерттеу объектісіне сәйкес келетін физикалық немесе абстракциялық жүйе. Модельдеу теориясына көбіне абстрактылық модельдер – объектінің белгілі бір тілдегі сипаттамасын қолданылады [19]. Модельдің абстрактылығы модельдің құраушы бөліктері физикалық элементтер емес, көбінесе қатынастар түріндегі асбтракциялық модель математикалық модель деп аталады. Математикалық модель $Y = F(X)$, түріндегі функционалдық тәуелділік формасында болады, мұндағы $Y = F(X)$, мұндағы $Y = \{y_1,...,y_m\}$ и $X = \{x_1,...,x_n\}$ -модельденетін жүйенің сәйкесті сипаттамалары мен параметрлдері, ал Ғ модель туындайтын функция [20]. Модельді құрастыру Ғ функциясын анықтау және оны $Y = F(X)$ мәндерін есептеуге қолайлы түрге келтіру болып табылады. Модель Y сипаттамаларын берілген X параметріне сәйкес бағалауға мүмкіндік береді және оңтайландыру әдістерін қолдана отырып, параметрлердің мәндерін таңдауға жағдай жасайды.

Модель жүйенің қасиеттерінің барлық жиынын немесе бір бөлігін қайталай алуы мүмкін. Жүйенің қасиеттерін құрамын зерттеу мақсатына байланысты жинақталып, модельді құрғанда төмендегі шамаларды анықтайды.

Зерттелетін $Y = \{y_1,...,y_m\}$ сипаттамалардың құрамын: Y сипаттамалары әсер ететін $X = \{x_1,...,x_n\}$ параметрлерінің құрамын;

$x_j \in x_j^*$, $j = 1,...,n-$ параметрлерінің өзгеру аралығын, n – модельдің анықталу облысы;

Дәлдікті модельдің негізінде Y сипаттамаларының бағалауының рұқсат етілген қажеттілігін:

Y – сипаттамаларының құрамы – жүйенің зерттелетін қаситеттеріне байланысты анықталады - өнімділігі, сенімділігі, құны және басқа қасиеттері, осы қасиеттерінің толықтай бейнеленуін қамтамасыз етуі қажет. X параметрлерінің құрамы жүйенің қалыптасуының барлық негізгі

бағыттарын қамтуы керек, оларды зерттеу модель көмегімен қайталанатын зерттеу мақсатын жұмыс істеу сапасына әсер етеді.

Жүйенің қалыптасуының зерттелетін нұсқауының шекарасы модельді қолдану аймағын анықтайды. Сипаттамалар мен параметрлер көп модельдің қолдану аймағы кең болған сайын модельді қолданып шешетін есептерге байланысты модель әмбебап болып келеді [20].

Сипаттамаларды бағалаудың рұқсат етілген қателіктері және параметрлердің берілу дәлдігі модельдің дәлдігіне қойылатын талаптарды анықтайды. Егер сипаттамалардың 10% өзгеріс жүйенің қайсыбір нұсқасын таңдап алуға маңызды болмаса, онда сипаттамалдарды анықтау дәлдігі ±5% болуы керек. Көптеген жағдайларда алдымен жұмыстық жүктемелердің параметрлері 20-25% салыстырмалы қателікпен жуықтап берілуі мүмкін. Бұл жағдайда модельдің жүйесінің сипаттамалары қателерді де 5-15% деңгейінде алу жеткілікті болады.

Сипаттамалар мен параметрлердің құрамына және анықтау аймағының барлық аймақтарында дәлдігіне қойылатын талаптарды қанағаттандыратын модель жүйеге сәйкес (адекватты) деп аталады.

Сәйкестілікке модельдің анықталу аймағы елеулі ықпал етеді. Іс жүзінде кез келген модель $X = \{x_1,...,x_n\}$ нүктесінің кішкене маңайында сипаттамаларды қайталауды жоғары дәлдігін қамтамасыз ете алады. Модель анықтау аймағы кең болған сайын жүйеге сәйкес болу ықтималдылығы аз болады.

Модельдің келесі қасиеті – күрделілігі. Модельдің күрделілігі екі көрсеткішпен сипаттау алынған: сипаттамаларды анықтаумен байланысты өлшемдік сәйкестікті және есептеу күрделілігі. Модельдің өлшемдік сәйкестігі – модельде параметрлері мен сипаттамалардың көрінісі.

$Y = F(X)$, сипаттамаларын есептеу кезіндегі есептеу күрделілігі F операторын бір рет жүзеге асыру кезінде орындалатын операциялар санымен анықталады [12]. Күнделікті есептеу күрделілігі ЭВМ ресурстарын пайдалану және процессорлық операциялары мен модельге қатысты ақпаратты сақтау үшін қажетті жады сыйымдылығына байланысты.

Есептеу күрделілігі модельдің өлшемділігімен байланысты өсетін монотонды функция. Сондықтан күрделі модельге бір мезгіде үлкен өлшемділік және есептеу күрделілігі тән.

Сонымен қатар модельдің күрделілігі модельденетін жүйенің күрделілігімен, модельдеу мақсаты (модель қайталайтын сипаттамалар мен параметрлер құрамы) қолдану аймағының өлшемдерімен және модельдің дәлдігімен анықталады. Жүйе күрделі болған сайын, яғни оның құрамына кіретін жұмыс істеуге әсер ететін элементтер мен үрдістердің саны көп болған сайын модель күрделене түседі. Қайталанатын сипаттамалар мен параметрлер санының, анықтау аймағы мен сипаттамалардың бағалау дәлдігінің өсуі модельдің күрделілігінің өсуіне әкеледі.

Объектілердің модельдері екі үлкен классқа бөлінеді: материалдық (физикалық) және абстракциялық (математикалық). Физикалық модельдердің ішінде аналогты модельдер кеңінен тараған. Математиканың дамуына байланысты математикалық модельдер кең таралды. Іс жүзінде математика түгелдей объектілер мен үрдістер модельдерін жасауға және зерттеуге бағытталған.

Математикалық модель жасау негізінен екі мақсатқа көзделеді:

- жүйенің құрылымын және жұмыс істеуін бір мәнді түсіну үшін формальді сипаттау;
- жүйені аналитикалық зерттеу үшін жұмыс істеу үрдісін көз алдына елестету.

Физикалық құбылыстардың математикалық моделін зерттеу үшін сандық әдістер қолданылады. Олар зерттелетін құбылыстың модельдейтін компьютерлік бағдарлама болуын керек етеді [22].

Математикалық модель жасау үшін жүйенің күйлерінің жиынын, сыртқы әсерлер (кіріс сигналдар), жауап (шығыс сигналдары), сол сияқты шығыс сигналдарымен сыртқы әсерлерін, олар ішкі күймен арақатынасын анықтау қажет. Олар жүйенің параметрлері, бастапқы шарттары және сыртқы әсерлерін бере отырып әр түрлі жағдайлардың көп санын зерттеуге мүмкіндік береді. Іздеп отырған функция модельдеудің нәтижесін кесте немесе графиктік түрде береді. Дифференциалдық теңдеудің сандық шешуін қарастырайық.

y=f(x) функциясының туындысы функцияның $\Delta y = f(x+\Delta x)-f(x)$ өсімшесінің Δx артуымен өсімшесі қатынасының аргументтің өсімшесі нольге ұмтылғандағы шегіне тең [23]:

$$y' = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \frac{\lim_{\Delta x \to 0} \left(f(x+\Delta x) - f(x) \right)}{\Delta x} = \frac{dx}{dy} \qquad (1.1)$$

Мұнда, dx – дифференциалы-аргументтің ақырсыз өсімшесі, ал dy – оған сәйкесті функция өсімшесінің сызықтық бөлігі. Туынды функциясының өсуінің шапшаңдығын және берілген *x* нүктесіндегі графиктік тіктігін көрсетеді. Ол жанама мен *x* осінің оң бағыты арасындағы бұрыштың тангенсіне тең.

Көп айнымалының $f(x_1...x_п)$ функциясының $f(x_1...x_п)$ нүктесінде x_1 нүктесіндегі дербес туындысы $f(x_1+\Delta x_1...x_п)$ өсімшесінің басқа тәуелсіз айнымалылар тұрақты деп есептегендегі Δx_1 өсімшесіне қатынасына шегі болып табылады.

Туындыны сандық есептеу үшін торлар әдісі қолданылады: $f(x_1...x_п\tau)$ функциясының өзгеру облысы бір өлшемді немесе кеңістік-уақыт торын құрайтын түйіндердің шекті санымен алмастырады. Үздіксіз аргументтің функциясынан дискретті аргументтің функциясына көшеді, әр түрлі уақыт қабатында есептеп, туындылар мен интегралдарды табады. Бұл кезде

y=f(x_1...$x_п$τ) шексіз кішкене өсімшелері және оның аргументтерінің өсімшелері кішкене, бірақ шектелген айырмаларымен алмастырылады.

y=f(x) функциясы берілсін. Интервалды а-дан в-ға дейін ұзындығы h=Δx элементар кесінділерге бөліп, x_1=a+iΔx, i=1,2…N тордың түйіндерінің шектелген жиынын аламыз, N – түйін саны. Бұл кезде Ω үздіксіз облысынан $\Omega_{\Delta x}$ торына, үздіксіз аргумент функциясынан $y_i=y(x_i)$ дискретті аргумент функциясына көшеміз. Ол үшін Тейлор қатарын жазамыз [24]

$$y(x) = y(x_i) + y_i^{'}\frac{(x-x_i)}{1!} + y_i^{''}\frac{(x-x_i)^2}{2!} + y_i^{(k)}\frac{(x-x_i)^k}{k!}, \quad (1.2)$$

$$y(x_{i+1}) = y(x_i) + y^{'}(x_i)\frac{h}{1!} + y^{''}(x_i)\frac{h^2}{2!} + ... + y^{(k)}(x_i)\frac{h^k}{k!}, \quad (1.3)$$

Мұндағы, $h=\Delta x=x-x_i$. Теңдіктің оң жағындағы қосындының алғашқы екі қосылғышымен шектесіп, $y(x_{i+i})=y(x_i)$ Δx жуық теңдігін аламыз. Ол функция анықтамасынан да шығады:

$$y^{'} = \frac{dy}{dx} = \lim_{\Delta x\to 0}\frac{\Delta y}{\Delta x} \approx \frac{\Delta y}{\Delta x} = \frac{y_{i+1}-y_i}{\Delta x},\ y_{i+1} = y_i + y^{'}(x_i)\Delta x\,. \quad (1.4)$$

Координатасы X_i нүктесінде сол, оң жақ және орталық айырмашылық туындылары:

$$y^{'}(x_i)_{-} = \frac{y(x_i)-y(x_{i-1})}{\Delta x},\ y^{'}(x_i)_{+} = \frac{y(x_{i+1})-y(x_i)}{\Delta x},\ y^{'}(x_i) = \frac{y(x_{i+1})-y(x_{i-1})}{\Delta x}. \quad (1.5)$$

Екінші туынды үшін:

$$y^{''}(x_i) = \frac{y(x_{i+1})-y(x_i)}{\Delta x} = \frac{y(x_{i+1})-y(x_i)}{\Delta x^2} - \frac{y(x_i)-y(x_{i-1})}{\Delta x^2} = \frac{y(x_{i+1})-2y(x_i)+y(x_{i-1})}{\Delta x^2}. \quad (1.6)$$

Тор қадамы аз болған сайын, табылған туындылардың дәлдігі жоғары болады [25]

Интеграл шексіз көп ақырсыз кішкене шамаларды, қосындысы. Ол сан жағын интегралданатын y=f(x) функциясының графигімен а және b интегралдау шектермен шектелген қисық сызықты трапецияның ауданына тең. Бұл қисық сызықты трапецияны ені Δx=h=(b-a)/ NV ұзындығы y_i=y(a+i Δx) болатын N жолақтарға бөлеміз. Жолақтың элементар ауданы $\Delta S=y(x_i)\Delta x$). Онда ізделіп отырған аудан төмендегі қосындыға тең:

$$S = \lim_{N\to\infty}\sum_{i=1}^{N} y(x_i)\Delta x = \int_a^b y(x)dx\,. \quad (1.7)$$

Интегралдың сан мәнін табу үшін:

$$S = \sum_{i=1}^{N} y(x_i)\Delta x\,. \quad (1.8)$$

қосындысын табу жеткілікті.

h-қадамы кіші, жолақ саны N көп болған сайын интегралдың табылған мәні дәлірек болады. Бұл әдіс тікбұрыштар әдісі деп аталады [26]. Дәлірек болып келетін трапеция әдісіне әрбәр і- жолақ биіктігі h= Δx, табандары y_2=y(a+i Δx), және y_{i+1}=y(a+(i+1) Δx) трапециямен

алмастырылады, сондықтан жолақтың ауданы $\Delta S_i=(y_i+y_{i+1}/\ \Delta x/2$. Интегралдың мәні осы трапеция тәрізді жолақтардың элементар аудандарының қоындысына тең:

$$S=\sum_{i=1}^{N}\frac{y(x_i)+y(x_{i+1})}{2}\Delta x\,. \qquad (1.9)$$

Y=y(x) қисығының астындағы қисықсызықты трапецияның ауданын табудың Монте-Карло[27] әдісі төмендегідей.

Берілген қисық сызықты трапеция ішінде болатын, интегралдау а және b шектерімен, Ох осімен және y=c горизонталіне шектелген тікбұрышты алайық. Тікбұрыштың ауданы (b-a)c координаталары (x_i,y_i) кездейсоқ берілетін N нүктесі тікбұрыштың ішіне орналастырайық. Қисық сызықты трапецияның ішінде орналасқан, яғни $y_i\langle y(x_i)$ шартын қанағаттандыратын нүктелердің п санын анықтайық. Қисық сызықты трапецияның ауданы п N неше есе кем болса, тікбұрыштың ауданына сонша есе кем болады. Сондықтан $N\to\infty$ болғанда n(b-a) c/N бөлшегі ізделінетін интегралға ұмтылады:

$$S=\int_a^b y(x)dx=\lim_{N\to\infty}\frac{n(b-a)c}{N}\,. \qquad (1.10)$$

Көптеген физика есептері бірінші ретті дифференциалдық теңдеулерге келеді. X(τ)-f(x(τ),τ)=0 [28). Ол шекті-айырмалы теңдеуге келтіріліп шешіледі:

$$\frac{x^{t+1}-x^t}{\Delta\tau}=f(x^t,t),\ x^{t+1}=x^t+f(x^t,t)\Delta\tau\,. \qquad (1.11)$$

$a\ddot{x}(\tau)+b\dot{x}(\tau)+kx(\tau)-f(t)=0$ дифференциалдық теңдеуі бірінші ретті дифференциалдық теңдеуге келтіріледі

$$\dot{x}(\tau)-v(\tau)=0,\ a\dot{v}(\tau)+bv(\tau)+kx(\tau)-f(\tau)=0\,. \qquad (1.12)$$

Дербес туындылы сызық дифференциалды теңдеу деп төмендегі қатынастарды айтады:

$$a_{11}\frac{\partial^2 f}{\partial x^2}+a_{22}\frac{\partial^2 f}{\partial y^2}+\ldots+b_{12}\frac{\partial^2 f}{\partial x\partial y}+b_{13}\frac{\partial^2 f}{\partial x\partial z}+\ldots+c_1\frac{\partial f}{\partial x}+\ldots+cf+\varphi(x,y,\ldots)=0\,. \qquad (1.13)$$

a_{11}, a_{22}..., b_{12}, b_{13}…c_1, c_2 коэффициенттері x,y немесе f(x,y,z) функция мәндеріне тәуелді болса, теңдеу сызықтық емес.

Егер y(x,y,z)=0 болса, теңдеу біртекті деп аталады. [14;29]

1. Гиперболалық теңдеулерге, бір өлшемді (ішек), екі өлшемді (мембрана) және үш өлшемді серпімді ортадағы тербеліс теңдеулері жатады:

$$\frac{\partial^2\xi}{\partial x^2}+\frac{\partial^2\xi}{\partial y^2}+\frac{\partial^2\xi}{\partial z^2}=\frac{1}{v^2}\frac{\partial^2\xi}{\partial\tau^2}+\beta\frac{\partial\xi}{\partial\tau}-F(x,\tau),\ \frac{\partial^2\xi}{\partial x^2}+\frac{\partial^2\xi}{\partial y^2}=\frac{1}{v^2}\frac{\partial^2\xi}{\partial\tau^2}\,. \qquad (1.14)$$

Телеграфтық теңдеулер (электромагнитті) тербеліс теңдеулері:

$$\frac{\partial u}{\partial x}+L\frac{\partial i}{\partial t}+Ri=0,\ \frac{\partial i}{\partial x}+C\frac{\partial u}{\partial t}+\frac{1}{R}u=0\,. \qquad (1.15)$$

Ішектің (мембрана) үшін шектік шарттар: 1) ішектің бір не бірнеше нүктесін тербелу режимі ξ(o;τ)=μ(τ); 2)бір не бірнеше нүктенің жылдамдығы: $\frac{d\xi(0,\tau)}{d\tau} = v(\tau)$; 3) бір не бірнеше нүктеге әсер ететін $\frac{d^2\xi}{d\tau^2} = F(\tau)$ берілген.

2. Параболалық теңдеулердің мысалдары: жылуөткізгіштік (диффузия, тұтқырлық) теңдеулері:

$$c\rho\frac{dT}{d\tau} = \frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right) + q(x,y,\tau). \quad (1.16)$$

3. Эллипстік теңдеулерге стационар а) электр өрісінің потенциалы үшін теңдеу жатады

$$\frac{\partial}{\partial x}\left(\varepsilon_x\frac{\partial\varphi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\varepsilon_y\frac{\partial\varphi}{\partial y}\right) = -\rho(x,y), \quad (1.17)$$

б) біркелкі емес қыздырылған пластинаның температурасын стационар таралуының теңдеуі

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right) = 0, \quad (1.18)$$

в)көздері жоқ сығылмайтын сұйық потенциалды ағысы теңдеу

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = 0. \quad (1.19)$$

Физиканың көптеген есептері Дирихиле [30;31;32] есебіне келтіріледі: ізделетін функцияның $\varphi(\Omega_0)$ облыс шекарасында мәні белгілі болғанда Ω тұйық облысында дербес туындымен берілген дифференциалдық теңдеуді шешу керек. Жылу өткізгіштіктің теңдеуі шешуге байланыс стационар емес есепті қарастырайық:

$$\frac{\partial T}{\partial\tau} = a\frac{\partial^2 T}{\partial x^2} + b\frac{\partial^2 T}{\partial y^2} + \frac{q(x,y)}{c\rho}. \quad (1.20)$$

$\Omega_{\Delta x;\Delta y;\Delta z}$ – торын енгізіп, шектік айырмаларға көшеміз

$$T_{i,j}^{t+1} = T_{i,j}^{t} + \left(a\frac{T_{i-1,j}^{t} - 2T_{i,j}^{t} + T_{i+1,j}^{t}}{\Delta x^2} + b\frac{T_{i-1,j}^{t} - 2T_{i,j}^{t} + T_{i+1,j}^{t}}{\Delta y^2} + \frac{q_{i,j}}{c\rho}\right)\Delta\tau. \quad (1.21)$$

Функция $T^{l}_{i,j}$ – уақыт қабатындағы мәнін біле отырып, оның келесі t+1 қабатындағы мәнін $T^{l+1}_{i,j}$ мәнін есептеу болады.

Стационар есептің мысалы ретінде Пуассон теңдеуін [11,15,22,24,32] шешуін талдайық:

$$\Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = f(x,y). \quad (1.22)$$

Теңдеу шектік айырмалар ретінде жазайық:

$$\frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{\Delta x^2} + \frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{\Delta y^2} = f(x_i, y_i),\ \Delta x=\Delta y=h. \quad (1.23)$$

Осыдан

$$\varphi_{i,j} = \frac{\varphi_{i+1,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} - f(x_i, y_j)h^2}{4}. \qquad (1.24)$$

Есепті шешуде біртіндеп жуықтау әдісі [17,28,33] қолданылады: тордың барлық ішкі түйіндерінде берілген функцияның кез-келген бастапқы мәні $\varphi_0(x_i,y_i)$ (нольдік жуықтау) беріледі; сыртқы түйінде функцияның мәні шекаралық шарттарға сәйкес болуы керек. Бірінші интерация жасалады: тордың барлық ішкі түйіндері қарастырылып, бастапқы шарттарға сәйкес функция жаңа $\varphi(x_i,y_i)$ дәлірек мәндері анықталады. Сосын екінші, үшінші,...және т.с.с. интерациялар жасалады, k-ншы интерация нәтижесі
(к+1)-інші интерация үшін бастапқы мән ретінде пайдаланылады.

Математикалық модельдер жасаудың бірыңғай әдістемесі жоқ. Бұл жүйелер класстарының көптеген түрімен байланысты [34]:

- статикалық және динамикалық;
- құрылымды немесе бағдарламалық басқару;
- тұрақты немесе айнымалы құрылым;
- тұрақты (қатаң) не ауыспалы (икемді) бағдарламалық басқару.

Кіріс әсер және ішкі күйінің сипатына байланысты жүйелер:

- үздіксіз және дискретті;
- сызықтық және сызықтық емес;
- стационар және стационар емес;
- детерминизацияланған және стохасимкалық болып бөлінеді.

жүйенің статикасын (құрылым бөліктерін және байланыс структурасын) бейнелейтін модельдер жасау онша қиын болмайды.
Динамикалық жүйе үшін статиканы жүйенің жұмыс істеуін сипаттаумен толықтыру керек.

Компьютерлік модельдеу технологиясы күрделі жүйелердің жұмыс жасауын зерттеуде ЭВМ іс жүзінде тиімді орындаудың мүмкіндігін қамтамасыз ету мағынасында бағытталған іс-қимылдардың негізі болып табылады. Оның көмегімен зерттеушінің іс-қимылдары модельмен жұмыс істеу барысында, пән облысын зерттеу мен модельден проблемалы ситуациядан бастап, жүйенің талдау компьютерлік тәжірибелерді құрып, жүзеге асыруға дейін ұйымдастырылады.
Модельдеу технологиясы туралы айтқанда, екі маңызды құраушысын атап өту керек [7;8;10]

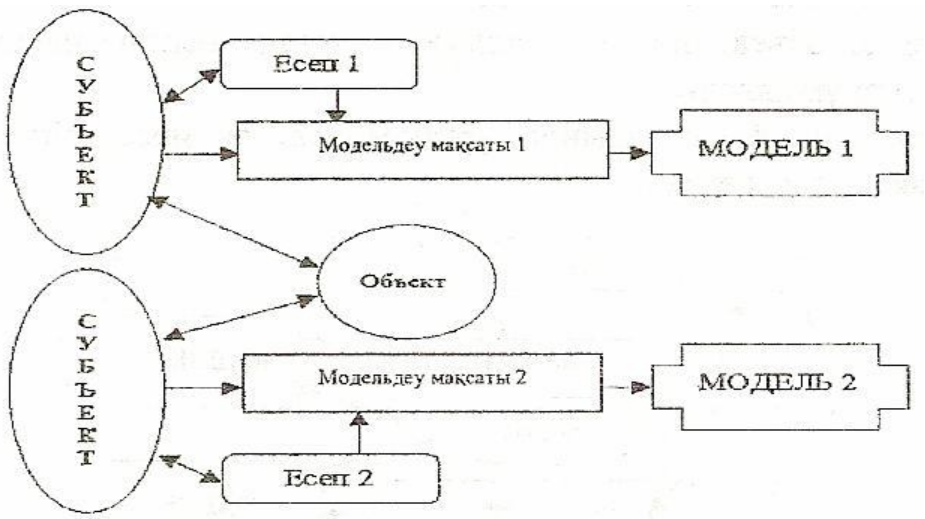
1. жүйелерді (объектілерді) ЭВМ көмегімен модельдеу барынша тиімді және үнемді әдістерін іс-жүзінде қолданудың заңдылықтарын көрсететін ғылым ретінде технологияның методологиялық құрама бөлігі,

2. күрделі объектілерді компьютерлік модельдеу барысында білім , шеберлік, өнер ретінде технологияның мақсаттары мен қолданбалы мақсаттары.

1.3 Компьютерлік модельдеудің негіздері
Модель құрудың негізгі этаптары. Модель түрлері

Модель құруды неден бастау қажет?

Біріншіден модельдеу мақсаты тұрғысынан объектіні талдау қажет. Бұл сатыда объектінің модельдеу субъектісіне таныс барлық қасиеттері қарастырылады. Объектінің көптеген қасиеттері мен белгілерінің
арасынан модельдеуде бейнеленуге тиісті қасиеттердің нақты болуы мүмкін
1.2 –сурет.



1.2 -сурет. Бірнеше субъектілердің бір объектіге түрлі модельдер құру мүмкіндігін көрсететін схема.

Модельдеу мақсаты анықталған соң - моделъдеу максаты тұрғысынан модельдеуші объектінің нақты белгілерін айқындау қажет.
Бұл белгілердің:

- объектінің сыртқы түріне;
- объектінің құрылымына;
- объектінің күйіне қатысы болуы мүмкін.

Модельдеу мақсатының түрліше болуына байланысты барлық жағдайлар үшін белгілерді, қасиеттерді, қатынастарды ерекшелейтін бірдей белгіленген тәсіл қазір жоқ.

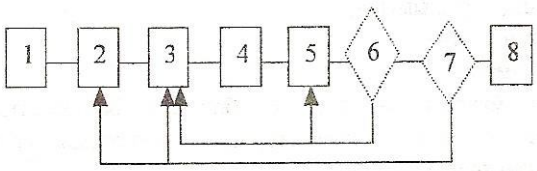
Нақты белгілердің дұрыс және толық. Ерекшеленуі құрылған модельдеудің берілген мақсатына сәйкестенеді, яғни оның модельдеу мақсатына адекваттылығына тәуелді болады. Модельдің адекваттылығы модельдеу объектісінде нақты ерекшеленген белгілердің қандай да бір формада бейнеленуіне тәуелді болады. Адекваттылық – модельдеудің негізгі түсініктерінің бірі.

Модельдеу объектісінің ерекшеленген белгілерін ұсыну формаларын таңдау – модельдеу практикасының келесі сатысы болып табылады. Модельдерді ұсынуда формалаудың сөздік сипаттама, сызба, кесте, формула, схема, алгоритм, компьютерлік бағдарлама сияқты түрлерінің қолданылуы мүмкін.

Объектінің ерекшеленген қасиеттері мен белгілерінің бейнелеу формасы тандалылған соң, таңдалған формадағы ерекшеленген қасиеттерге байланысты формалдау жұмысына кірісу қажет.

Формалдау процесі, мысалы математикалық модель бұйымның жиналу сызбасын құруда өзіндік ерекшеліктері мен сатыларына иелік етеді.

Формалдау сатысының нәтижесі ақпараттық модель болады. Модельдеу процесін аяқтау туралы айтпас бұрын құрылған модельдің модельдеу мақсатына және объектіге адекваттылығын тексеру қажет. Құрылған модельде мақсатқа сәйкес қайшылықтар кездессе сызбаны түзету, бағдарламаға өзгерістер енгізу, қолданылатын формулаларды нақтылау әрекеттерін орындап, модельдің дәлдігін қайта тексеру қажет. Алынған модельдің модельдеу объектісінің бейнелену адекваттылығына талдау жасап, модельдеу мақсатына жету – модельдеудің соңғы сатысы. Модельдеу сатыларының арасындағы өзара байланысы 1.3-суретте көрсетілген.



1.3-сурет. Модельдеу сатыларының арасындағы байланысты көрсету схемасы.

Қазіргі кезде әрбір жағдайда объектінің қандай белгілі қасиеттерінің нақты қасиет ретінде қарастырылатыны туралы әмбебап анықтамалық ереже жоқ.

Модельдеу шарты мүмкіндік берсе түрлі қасиеттерінің құрамымен бірнеше модельдер құрып, олардың объектіні модельдеу мақсатына адекваттылығын бағалау қажет.

Формалдау - модельдеу объектісінің нақты қасиеттері мен

белгілерін таңдалған формаға келтіру.

Ақпараттық модельдерді бейнелеу формасының сөздік сипаттама, кесте, сурет, алгоритм, сызба түрінде болуы мүмкін.

Модель түрлері

Барлық модельдердің көпбейнелілігі негізінен үш топқа бөлінеді:

- материалды (табиғи) моделъдеуші объектінің сыртқы түрін, құрылымын (кристал торлардың модельдері, глобус), жағдайын (самолеттің радио басқарылымды моделі) бейнелейтін кішірейтілген немес ұлғайтылған көшірмелері;
- бейнеленуші модельдер (геометриялық нүктелер, математикалық маятник, идеал газ, шексіздік);
- ақпараттық модельдер - модельденуші объектінің ақпаратты кодтау тілдерінің бірінде жазылған сипаттамасы (сөздік сипаттау, схемалар, сызбалар, картиналар, суреттер, ғылыми формулалар, бағдарламалар).

Информатика курсында негізінен ақпараттық модельдер қарастырылады. Субъектінің практикалық қызметінің сферасы модельдеу объектісін басқару процесіндегі модельдің қатысына байланысты нақтылануы мүмкін. Бұл жағдайда модельдің келесі түрлері: тіркелуші, эталондық, болжамдық, оңтайланған, имитациялық деп бөлінеді.

Модельдеу құбылысының қосымша мүмкіндіктерін ашуға мүмкіндік беретін модельдер кластарының басқалай да бөліну түрлерін таңдауға болады 1.4 -сурет.

Ақпараттық модель

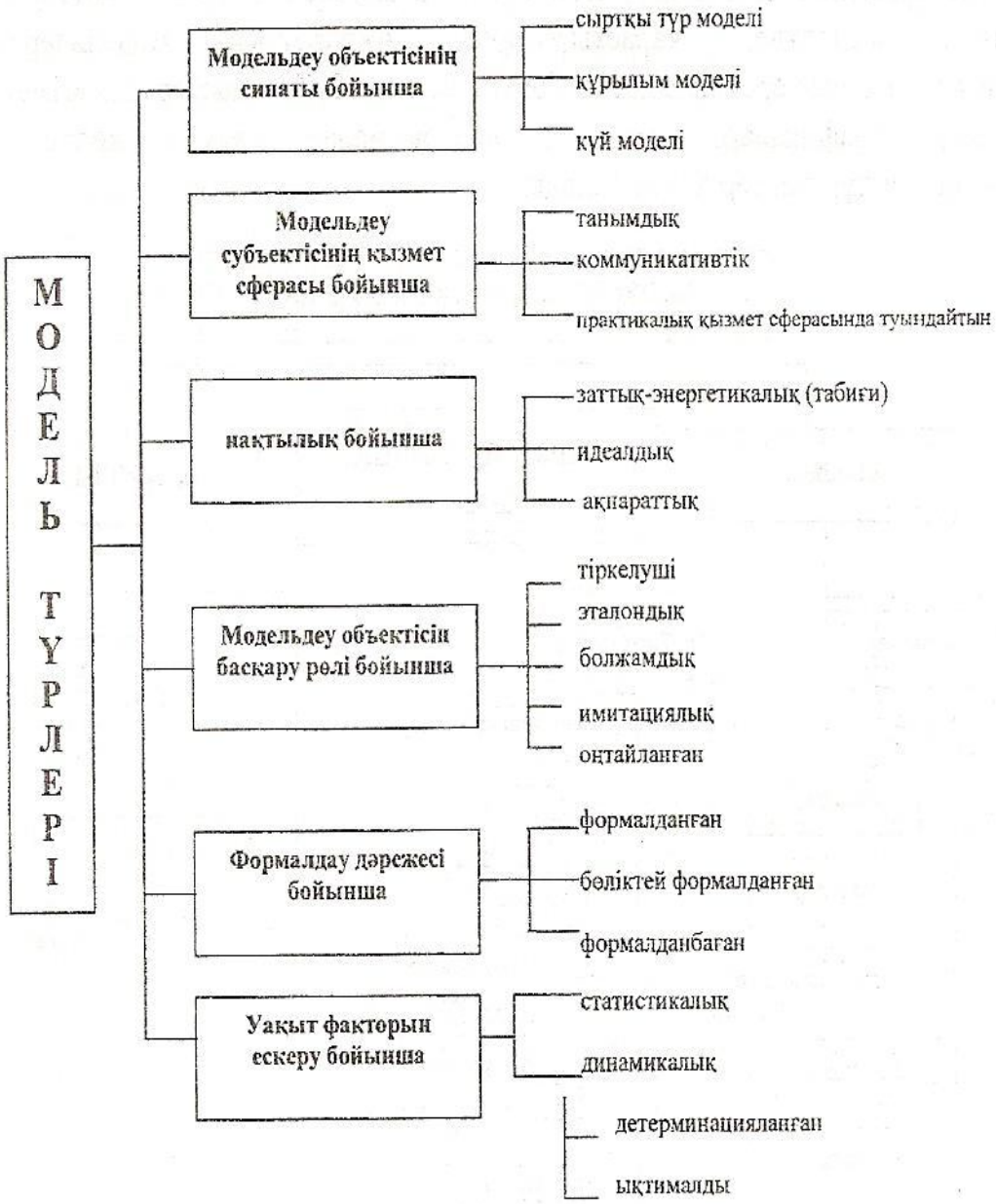
Ақпараттық модель (Информационная модель; infortation todel)

1) басқару жүйесінде - автоматтандырылған өндеуге жататын ақпарат айналымының процесін параметрлік ұсыну;
2) мәліметтер базасында тұтастық шектеулер жиынтығы мәліметтер құрылымын тудыратын ережелердің олармен жургізілетін операциялардың, сондай-ақ рұқсат етілетін байланыстар мен мәліметтердің мәнін, олардың өзгерістерінің тізбегін анықтайды; мәліметтермен олардың арасындағы қатынастарды математикалық және программалық тәсілдермен ұсыну; ақпараттық құрылымдар мен олармен жүргізілетін операцияларды формалдық баяндау [l8,20].

Ақпараттық моделъдердің басқада ақпарат түрлері сияқты өзіндік тасымалдаушысы болуы керек. Олар қағаз, сынып тақтасы, бейнелеуге болатындай кез-келген бет болуы мүмкін.

Бұл тасымалдаушыларда модельдер түрлі "физикалық" тәсілдермен: қалам, бор, бояу, диапроекторлық жарық бейнесі көмегімен жазылады.

1.4 -сурет. Модельдер классификациясы.



Моделъ (фр. modele, ит. modell0, лат. modulus - өлшем, үлгі) бұл:

- нақты объектінің қарапайымдандырылған ұқсасы;
- заттың кішірейтілген немесе ұлғайтылған түрдегі макеті;
- табиғат пен қоғамдағы қандай да бір процесстің құбылыстың бейнесі, сипаттамасы және схемасы;
- жұмыс істеуі анықталған параметрлер бойынша нақты объектінің жұмыс істеуіне ұқсас физикалық/ақпараттық аналогы;
- анықталған шарттарда тұпнұсқа объектінің бізді қызықтыратын қасиеттері мен сипаттамасын алмастыра алатын алмастырушы-объектісі;
- модельдеу мақсаты тұрғысынан оқып үйренетін

объектінің
/құбылыстың кейбір нақты жақтарын бейнелейтін жаңа объект.

Ақпараттық модель - модельденуші объектінің ақпаратты кодтау тілдерінің бірінде сипатталуы.

Модельдеу - бұл:

- нақты бар объектілердің (заттар, құбылыстар, процестер) модельдерін құру;
- нақты объектіні қолайлы көшірмемен алмастыру;
- таным объектілерін модельдері арқылы зерттеу.

Модельдеу кез-келген мақсатқа бағыттылған қызметтің ажырамас бөлігі. Модельдеу танымның негізгі әдістерінің бірі.

Нақты қызметтердегі объект модельдері төмендегі жағдайларға пайдаланылады:

- материалдық заттарды бейнелеу; белгілі фактілерді түсіндіру;
- болжамдар құру;
- зерттелінетін объект туралы жаңа білімдер алу;
- болжау;
- басқару және т.с.с.

Соңғы кездердегі ғылым мен ақпараттық технологиялардың қарыштай дамуы барлық дерлік ғылыми-зерттеу жұмыстарында зерттелетін объектіні модельдеу жұмыстарын өз деңгейінде жүргізуді талап етуде. Модельдер барлық жерде дерлік кездеседі. Олардың саны орасан зор. Олардың кейбірі ескіреді, ұмытылады, жоғалады (3-сурет).

Ақпараттарды модельдеу түрлерін таңдауда және құруда (1.5-сурет) зерттеушінің маман ретіндегі танымы мен біліктілік деңгейі, эстетикалық талғамы көрінеді. Дұрыс таңдалған және өз дәрежесінде тиімді құрылған модель түрлерін зерттеу жұмыстары жеңілдетіп, объект туралы толығырақ мәлімет алуға септігін тигізеді. Әрбір модель үшін оның кеңістіктегі "субъект-объект-нақтылық" орнын анықтауға болады.

Ақпараттық модель әркайсысын бейнелеуге таңдалған бейнелеу тілдерінің формалдылығын сипаттай алады. Әрбір ақпараттық модельді кеңістіктегі "субъект-объект-формалдау дәрежесі" нүктелеріне сәйкес қойып көруге болады.

Ақпараттық модельдерді сипаттау тілі бойынша мысалы, математикалық модельдер, кестелік модельдер сияқты мақсатты түрде классификациялауға болады.

Модельдеу тілі (simulation language) - зерттеліп жатқан объектіні үлгілеу үшін қажетті бастапқы ақпарат берілетін жобалау тілі [2l]. Информатика курсында компьютер көмегімен құруға, зерттеуге болатын модельдер қарастырылады. Ақпараттық модельдерді компьютерлік деп ерекше класқа бөлуге бола ма? Компьютер көмегімен мәтіндер, графикалар, кестелер, диаграммалар сияқты көптеген объектілерді құруға, зерттеуге болады.

1.5-сурет. Ақпараттық модельдер түрлері.

Компьютерлік моделъдер

Компьютерлік моделъдердің, ақпараттық моделъдерден сапалық айырмасы жоқ. Компьютерлік модельдеуді өзіндік ерекшеліктеріне орай ақпараттық модельдеудің ерекше түрі деп айтуға болады.

Компьютерлік модель (cotputer todel*)* - 1) таңдалынған бағдарламалық ортаға бейімделінген ақпараттық модельді ұсыну формасы; 2) бағдарламалық ортаның құралдарымен жасалынған модель [21].

Компьютерлік модельдерге байланысты бастапк:ы жұмыстар гидравлика, жылу алмасу, қатты дененің механикасы т.с.с

есептер тобын шешуде жүргізілді.

Модельдеу ЭЕМ мүмкіндіктері, жұмыс істеу принциптері мен математикалық модельдердің адаптациясы болатын күрделі тендеулер жүйесінің сандық шешімін бейнелейді. Физикадағы компьютерлік модельдердің табыстары химия, электроэнергетика, биология есептерін шешуде де кең таралды. Компьютерлік модельдеу негізінде шешілетін есептердің күрделілігі ЭЕМ-нің мүмкіндіктеріне байланысты шектеледі.

Күрделі жүйелерді талдаудағы компьютерлік модельдеу зерттелетін объектінің математикалық-логикалық күйін модельдеу, объектінің к:ызметтік алгоритміне айналатын, компьютерлерге арналған бағдарламаларды комплексті түрде дайындайтын имитациялық модельдеу болып табылады.

Кез-келген объект күйін имитациялауға болады, бірақ имитациялық модельдеу бәрінен бұрын таңдалған басқару стратегиясына тәуелді күрделі жүйелердің алдағы уақыттағы күйін болжаудың зерттелуін қарастырады.

Қазіргі кезде компьютперлік модель ретінде:

- өзара байланысты компьютерлік суреттердің, кестелердің, схемалардың, диаграммалардың, графиканың, анимациялық фрагменттердің, гипертекстердің көмегімен сипатталған объектінің шартты бейнесі айтылады. Бұл түрдегі компьютерлік модельдер құрылымдық-функционалдық деп аталады;

Түрлі факторлардағы объектіге әсер ету шарттарының фуцкциялану процесін имитациялауды реттелген есептеулер мен графикалық бейнелеулер нәтижесін шығаруға мүмкіндік беретін жеке бағдарламалар комплекстері аталады.

Мұндай модельдер имимтациялық компьютерлік модельдер деп аталады.

Имитациялық компьютерлік модельдеу модель бойынша модельдеуші жүйенің сандық және сапалық функциялану нәтижесін алуға негізделген. Модельдерді талдау нәтижесінде алынған сапалық қорытындылар күрделі жүйенің құрамы, даму динамикасы, орнықтылығы бүтіндігі сияқты бұрын белгісіз болып келген қасиеттерін ашуға мүмкіндік береді. Сандық қорытындылар негізінен жүйені сипаттайтын болашақ және бұрыннан белгілі параметрлердің мәндерін түсіндіруде болжамдық сипатты иеленеді.

Компьютерлік модельдеудің мақсаты - экономикалық, әлеуметтік, ұйымдастырушылық-техникалық сипатта шешімдайындап, қабылдауға пайлаланылуы мүмкін мәліметтер алу.

Төменде келтірілген анықтамалар модельдер мен олардың айырмашылық ерекшеліктерін нақтылай түсінуге көмектеседі.

Табиғи (физикалық, заттық-энергетикалық) модельдеу - модель мен модельдеуші объект өзара нақты объектілерді немесе бірдей/түрлі

физикалық процестерінің табиғатын бейнелейтін модельдеу.
Программалық модельдеу - 1) құрылғының немесе жүйенің іс- әрекетін программаның көмегімен модельдеу;
2)программалық жасақтаманың жұмысын модельдеу [22].

Ақпараттық модель - бұл объектінің қандай да бір тілдегі сипаттамасы. Модельдің абстракциялық компоненттері физикалық дене емес сигналдар мен белгілер болады. Түрлі белгілер жүйелерінде ақпараттық процестерді сипаттайтын белгілік модельдер класы.

Модуль ілінісіуі. Программа құрылымын дайындаудың тәсілдері

Модуль ілінісуі –бұл модульдің берілгендер бойынша басқа модульдерге тәуелділігінің өлшемі.

Берілгендердің берілу жолымен сипатталады. Модульдің басқа модульдермен ілінісуі әлсіз болған сайын оның тәуелсіздігі жоғары ілінісу деңгейін бағалау үшін Майерс реттелген модуль ілінісу түрлерінің алтауын ұсынады. Ең төмен деңгейлі ілінісу – бұл _мәні бо йы нша ілінісу.

Бір модульдің екінші модульдің құрамындағысын (константа, айнымалы т.б.) тікелей пайдалануы осыған жатады. Мұндай ілінісуге жол бермеу керек. Жалпы аймақ бойынша ілінісуді де қолдануға кеңес берілмейді. (бірнеше модульдің бір ғана жады аймағын пайдалануы). Қолдануға кеңес берілетін модуль ілінісуінің бір ғана түрі бар, ол – параметрлік ілінісу не берілгендер бойынша ілінісу. Бұл модульді шақырған кезде берілгендердің модуль параметрі ретінде берілетін жағдай немесе берілгендердің қайсыбір функцияны есептеу үшін басқа модульді шақыру нәтижесі ретінде берілетін жағдай.

Модуль ілінісуінің бұл түрі программалау тілдерінде процедураны (функцияны) шақыруда жүзеге асады.

Модул байланыстығы модульді алдыңғы шақыруларынан тәуелсіздігі. Модульді алдыңғы шақыруына байланыссыз деп атаймыз, егер модульді шақыру нәтижесі тек параметрінің мәніне ғана тәуелді болса (және алдыңғы шақыруларынан тәуелсіз болса). Модульді алдыңғы шақыруларына байланысты деп атаймыз, егер модульді шақырудың нәтижесі алдыңғы шақырылғанда өзгерген ішкі жағдайына тәуелді болса.
Осыған байланысты мына кеңестер беріледі:

- егер модульдердің төмен деңгейлі ілінісуге әкеп соқпайтын болса,
- онда алдыңғы шақыруына байланыссыз модульдерді қолдану керек;
- параметрлік ілінісуді қаматамасыз ету үшін қажет болған жағдайда ғана алдыңғы шақыруына байланысты модульдерді қолдану керек;
- алдыңғы шақыруына байланысты модульдің спецификациясында осы

байланыстылық айқын көрсетілсін.

Соның арқасында модульдің әр түрлі жағдайда шақыру нәтижесін болжау мүмкін болсын.

Жоғарыда айтылып өткендей, программаның модульді құрылымы ретінде ағаш түріндегі құрылымды (бұтақты ағашты да қоса) қолдану келісілген. Мұндай ағаштық түйіндерді программалық модульдер орналасады. Ал бағытталған доғалар (стрелкалар) модульдердің статикалық бағыныштылығын көрсетеді, яғни әрбір доға модуль текстінде (доғаның басында орналасқан модуль) келесі модульге (доғаның екінші ұшында орналасқан модуль) сілтеме бар екендігін білдіреді. Басқаша айтқанда, әрбір модуль өзіне бағынышты модульді шақыруы мүмкін, яғни осы модульдер арқылы өрнектелуі мүмкін. Сонымен бірге программаның модульді құрылымы программаны құрайтын модульдердің спецификациялар жиынтығын өз ішіне алу керек.

Программалық модульдің спецификациясы біріншіден, оның кірістерінің (входов) синтаксистік спецификациясынан (ол қолданып отырған программалау тілінде модульді синтаксистік дұрыс шақыруды құруға мүмкіндік береді), екіншіден модульдің функционалды спецификациясынан (осы модульдің әрбір кірісі бойынша орындалатын функциялардың семастикасын бейнелеу) тұрады.

Модульдің функционалды спецификациясы ПС-ң функциясының спецификациясы сияқты құрылады.

Программа құрып, дайындау процессінде оның модульдік құрылымы әртүрлі түзілуі мүмкін және осы құрылымда көрсетілген модульдерді отладка жасау және программалау ретін анықтауда пайдаланылуы мүмкін.

Жоғары қарай құру тәсі лі: Алдымен программаның ағаш түріндегі модульді құрылымы құрылады. Одан соң кезекпен модульдер ең төмен деңгейдегіден (ағаш жапырақтары) бастап программалана бастайды.

Әрбір программаланатын модуль үшін керек болатын модульдер программаланып қойылатындай кезекпен құрылады. Программаның барлық модульдері программаланып болған соң, жоғарыдағыдай кезекпен оларды тестілеу және отладка жасау жүргізіледі. Бір қарағанда программаның осындай ретпен құрылуы керек те сияқты: әрбір модуль программалану кезінде программаланып қойылған бағынышты модульдермен өрнектеле алады, және тестілеу кезінде отладка жасалынған модульдер дайын қолданылады. Айтылып жатқан тәсілдердің барлығы программа дайындалу барысында оның ағаш түрдегі құрылымның түйіндерін (модульдерін) қандай ретпен айналып өтуіне қарай әр түрге бөлінеді. (Айналып өту түрлері: жоғарыдан –төменге, оңнан–солға, т.б).Солардың бір түрі дәлірегі конструктивті жандасудың бір түрі мақсаттты – конструктивті жүзеге асыру. Бұл тәсілдің мәні мынадай: конструктивті жандасуды ұсатала отырып, алдымен

программаның ең қарапайым (ықшамды) варианты үшін кеерқті модульдер жүзеге асырылады [24].

Компьютерлік математикалық үлгілеу (модельдеу) кезеңдері мен мақсаттары

Сандық эксперименттерден тұратын компьютерлік математикалық үлгілеу үрдісін қарастырайық.

I –кезең .Үлгілеу мақсаттарын анықтау. Олардың ішіндегі ең негізгілері:

а) үлгі – нақты объектің қалай ораналасқаны, оның құрылымың негізгі қасиеттері, даму заңдары және қоршаған ортамен өзара әрекеті туралы түсіну үшін қажет (түсіну үлгілері);

б) үлгі – объекті (немесе үрдісті) қалай басқаруды үйрену үшін және берілген мақсаттар мен критерийлар барысында тиімді басқару әдістерін анықтау үшін қажет (басқару үлгілері);

в) үлгі – берілген тәсілдер мен объектпен өзара әрекеттесу формаларын жүзеге асырудың тура және жанама салдарын болжау үшін қажет (болжау үлгілері).

II – кезең. Үлгілеудің маңызды кезеңдерінің бірі болып шығыс шамаларының өзгеруіне әрекет ететін маңыздылық деңгейі боынша кіріс параметрлерін бөлу болып табылады. Мұндай үрдіс ранжирлен деп аталады. Үлгілеу объектісінің маңызы төменірек болатын факторларын елемей оны кесектендіруге болады, бұл үлгінің басты қасиеттері мен заңдылықтарына көбірек көңіл бөлуге көмектеседі.

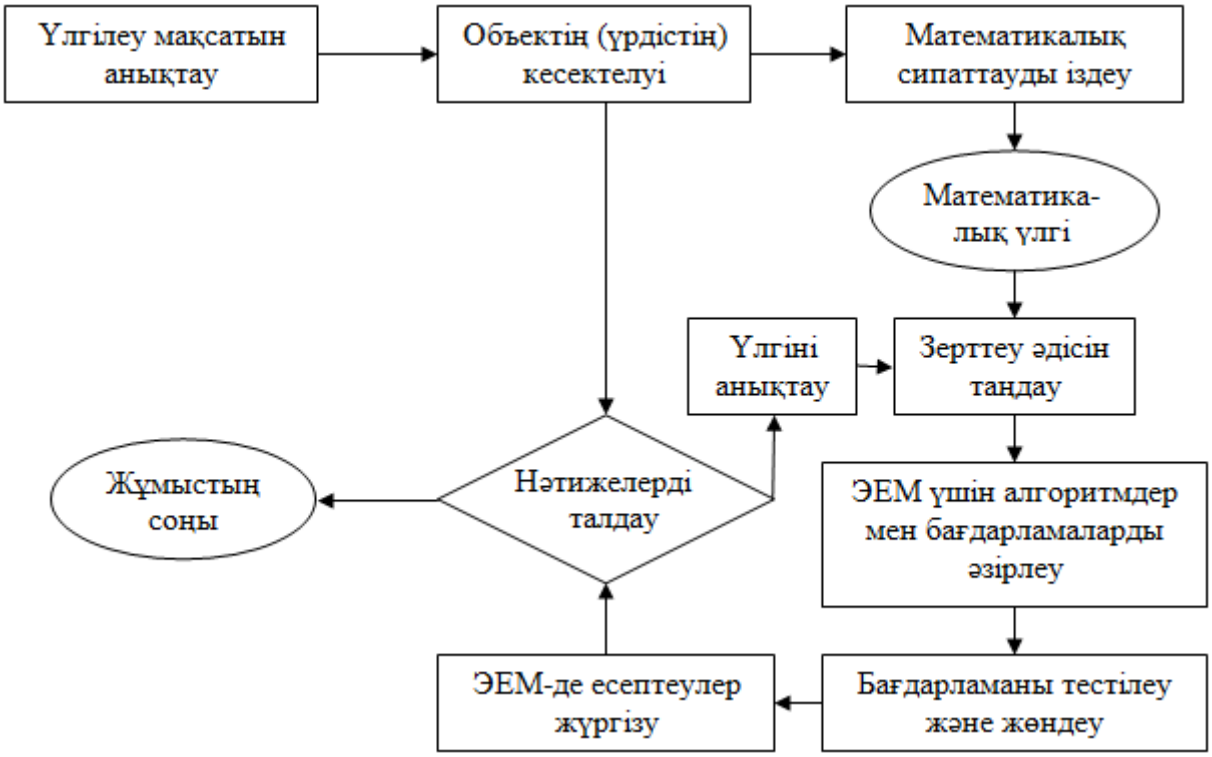
III – кезең . Математикалық сипаттауды іздеу . Бұл кезеңде үлгінің абсрактілі тұжырымынан нақты математикалық сипаттау тұратын тұжырымынға өту қажет. Бұл жағдайда үлгі теңдеу, теңдеулер жүйесі, теңсіздіктер жүйесі, дифференциалдық теңдеулер жүйесі және т.б. түрінде жасалады.

IV – кезең. Әдісті және зерттеуді таңдау. Әдісті іздеуде оның тиімділігін, тұрақтылығын және т.б. ескеру маңызды.

V – кезең. ЭЕМ үшін алгоритмдер мен бағдарламаларды әзірлеу. Бұл шығармашылық жағынан қиын қалыптасатын үрдіс.

VI – кезең. Нәтижелерді тестілеу және жөндеу. Тестілік тапсырмалар үшін бағдарламаны орындау қажет.

VII – кезең. Сандық эксперимент.ЭЕМ-де есептеулер жүргізу, нәтижелерді талдау, нақты үлгі адекваттылығын тексеру үлгісін анықтау, үлгіні анықтағаннан кейін баспатқы кезеңге қайту 1.6-сурет.



1.6-сурет.
Компьютерлік математикалық үлгілеу үрдісінің жалпы сызбасы

Математикалық модельдеу классификациясы

Математикалық үлгілеу классификациясы.

Үлгілеудің мақсаттарын құрайтын үлгілеудің жалпы заңдылығына негізделген үлгілер классификациясын берейік:

- дискриптивті (сипаттау) үлгілер;
- оптимизациялық үлгілер;
- көпкритерийлі үлгілер;
- ойын үлгілері;
- имитациялық үлгілер.

Дескриптивті үлгілер (ағ. descriptive - сипаттамалық) модель - объектінің қандай да бір тілде сөздік сипатталуы. Мысалы, Күн жүйесіне енген камета қозғалысын үлгілей отырып, біз алдымен оның ұшу траекториясын, Жерден қанша қашықтықта өтетінін және т.б. сипаттаймыз (болжаймыз), яғни, сипаттау мақсаттарын құрастырамыз.

Опитимизациялық үлгілер – үрдістердің басқа да деңгейлерінде қандай да бір мақсатқа жету үшін оларға өзара әрекеттесе аламыз. Есептің максималды пайдасына жету үшін өз әрекетімізге қатысты параметрлер таңдауға ұмтыламыз, яғни, үрдісті оптимизациялаймыз.

Көпкритерийлі үлгілер – үрдісті бірнеше параметрлер бойынша бірге оптимизациялауға тура келеді, бірақ олардың мақсаттары әртүрлі

болуы мүмкін. Онды үлгілеу барысында бірнеше критерийлер бірігуі мүмкін.

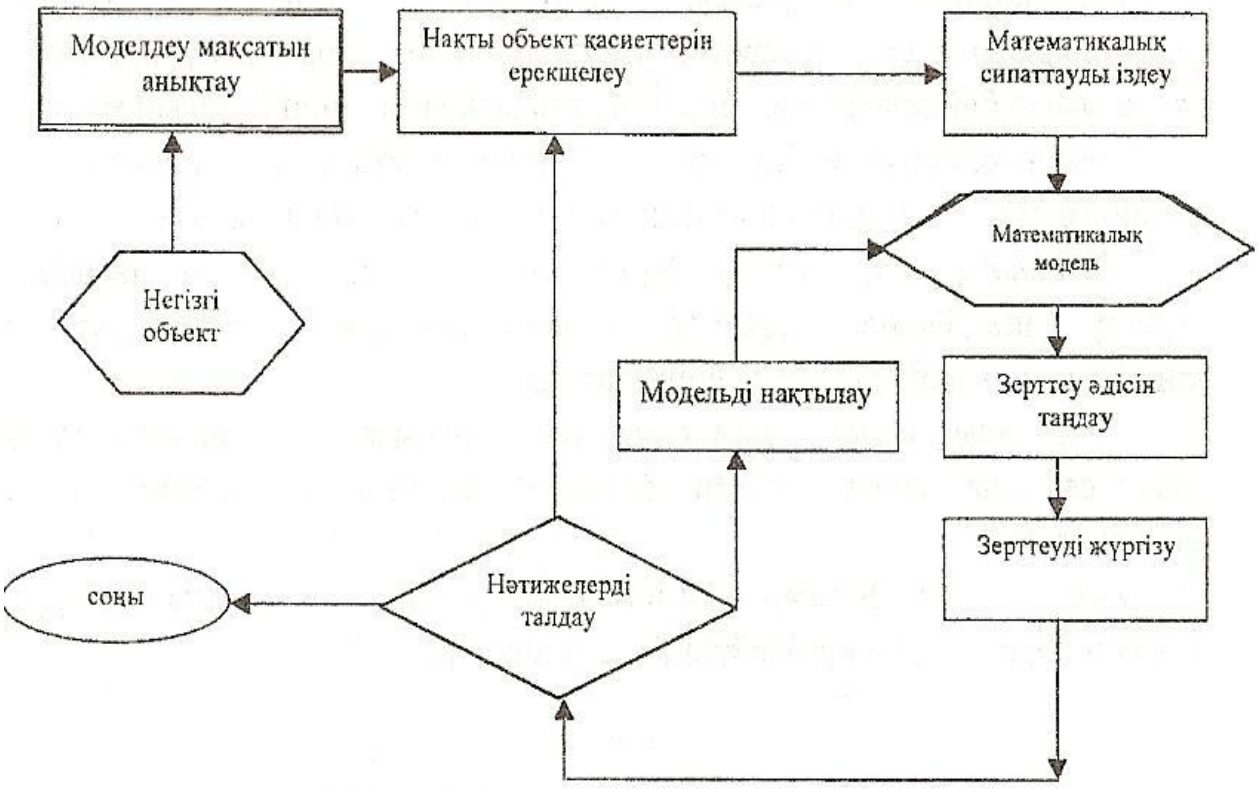
Ойын үлгілері бұл тек ойындарға ғана қатысты болмауы мүмкін. Қазіргі заманғы математикада айтарлықтай күрделі бөлім – жеткіліксіз ақпараттар шартындағы шешім қабылдау әдісін оқып-меңгеретін – ойын теориясы бар.

Имитациялық үлгілер – бұл объектке ұқсайтын үлгі, яғни нақты объектіні имитациялайтын үлгі. Имитациялық үлгілер – үлкен жүйе қасиетін оны құрайтын объектердің өте қарапайым әрі нақты қалыптасқан шарты барысында сипаттауда қолданылады. Онда математикалық сипаттау жүйенің макроскопиялық сипаттамасын табу барысында үлгілеу нәтижелерін статистикалық өңдеу деңгейінде жүргізіледі.

Математикалық модель:

- объект жене объекті элементтерінің қасиеттеріне, параметрлеріне, сыртқы әсерлердің күйін сипаттаумен анықталатын математикалық қатыстар (формулалар, теңсіздіктер, теңдеулер, логикалық қатыстар) тілінде жазылған жиынтық;

- математикалық символдар көмегімен өрнектелген объектінің жуық сипаттамасы (1.7-сурет). 1.7-сурет. Математикалық модельдеу процесінің жалпы схемасы



1.7-сурет. Математикалық модельдеу процесінің жалпы схемасы

Математикалық модель (matemathical situlatioп)- объектінің қызметі мен құрылымын сипаттайтын математикалық тәуелділіктер жүйесі, яғни математикалық формулалар мен теңдеулер арқылы өрнектелетін объектілердің математикалық сипаттамалары.

Математикалық модельдеу (matetathical todeliпg) - процестер мен құбылыстарды олардың математикалық модельдерінде зерттеу әдісі.
Тәжірибе жасауға мүмкіндік болмаған, қиын немесе қолайсыз жағдайларда ғана пайдаланылады. Жеке жағдайда аналитикалық модельдеу болып табылады [22,26].

Математикалық модельдер химия, биология, экология, гуманитарлық және әлеуметтік ғылым салалары үшін дәстүрлі модель түрі болып табылады.
Статистикалық модельдер уақыт мезетіне тәуелсіз жасалатын өзгерістерге орай объектілердегі тыныштық пен тепе-теңдік ткүйін бейнелейді. Бұл модельдерде уак:ыт параметрі болмайды.

Семантикалық модель (setaпtic model)- семантикалық жадта ұғымдарды граф түрінде ұсыну. Оның төбелерінде ұғымдар, терминалдық төбелерінде элементарлық ұғымдар орналасқан, ал доғалар ұғымдрдың арасындағы қатынастарды көрсетеді.
Семантикалық модельдеу sепаптic situlatioп)- іске асыруда тәуелсіздігін сақтауда мәліметтердің мазмұнын (жеке-жеке формальдық тәсілмен) барынша толық жеткізу әдістерін әзірлеу мен қолдану[l).

Дuнамикалық модель уақытқа байланысты объект күйін сипаттайды, яғни модельдер уақытқа байланысты объектіде өтетін процестерді бейнелейді. Дербес жағдайда функциялану және даму модельдерін айтуға болады.

Детерминацuялық модельдер- кездейсоқ әсерлер болмайтын процестерді бейнелейді.

Ықтималды модельдер - объектінің күйінің кездейсоқ ішкі/сыртқы әсерлермен анықталатын сипаттамасы. Ықтимал өзгеру сипатын уақытқа байланысты алдын-ала болжауы мүмкін емес процестер мен оқиғалардың сипаты.
Имитацuялық алгоритмдік модельдеу - объектінің кездейсоқ факторлардың әсерін ескеретін, уакыт бойынша формалану процесі мен құрылымын бейнелейтін алгоритм формасындағы сипаттамалық мазмұны.

Гносеологuялық модельдер - табиғатгың объективті заңдарын оқып үйренуге бағытталған (Күн жүйесі моделі, биосфераның дамуы т.с.с.).

Концептуалдық модель зерттелетін объектіге және анықталған

зерттеу шеңберіне қатысты себеп-салдарлық байланыстар мен заңдылықтарды айқындауды сипаттайды.

Сенсуалдық модельдер (лат. sensualis - сезімге, түйсікке негізделген) - адам сезіміне ықпал ететін сезімдік, эмоциялық (музыка, поэзия) модельдер.

Аналогтық модельдер - өзі нақты объект ретінде іс атқаратын, бірақ дәл сондай бейнеде көрінбейтін объект аналогы.

Модель қасиеттері.

Модель құру нәтижесінде анықталған қатынастарда негізгі объектімен сәйкес келетін жаңа объект құрылады. Жаңа объектінің модельдеу объектісі болу мүмкіндігі бар. Демек, әрбір объекті түрлі модельдерге ие. Нәтижесінде кейбірі басқа объектілердің модельдері болатын объектілердің шексіз жиыны алынады. Осы жиын мен оның элементтері арасындағы қасиеттерді қарастырамыз.

Теория жүзінде объектілер мен модельдер жиынында:

- объект пен басқа объект арасында;
- объект пен оның моделі арасында;
- объект пен басқа объект моделінің арасында;
- модель мен модельденуші объект арасында;
- модель мен басқа объект арасында;
- модель мен объектінің басқ.а моделінің арасында
- модель мен басқа объектінің моделінің арасында қатынастар болуы мүмкін.

Объект пен оның модельдері арасындағы қатынастары қасиеттер арқылы сипатталады.

Модельдің ең басты қасиеті модельдеу мақсатына байланысты кейбір қатынастардың модельдеу объектісіне ұқсастығы болып табыладыr.

Математикада ұқсастық қатынасы симметриялық, рефлексивтік, транзитивтік қасиеттермен өрнектеледі.

Симметриялық. Негізгі объектіні өз моделінің сәйкесті моделі ретінде қарастыруға болады.

Транзитивтік. Объект моделіне құрылған модель, негізгі объектінің моделі болады.

Рефлексивтік. Объект өзінің дәл моделі бола алады.

Кез-келген зерттеу объектісін кейбір қасиеттерде бір бүтінді бейнелейтін өзара байланысты элементтер тобынан құралған жүйе ретінде қарастыруға болады.

Бір жүйенің әрбір элементіне басқа жүйенің әрбір элементі сәйкестенетін (және керісінше) өзара бірмәнді сәйкестігі бар екі жүйе изоморфты деп аталады.

Бір жүйенің әрбір элементіне басқа жүйенің элементі сәйкестенетін (керісінше емес) өзара бірмәнді сәйкестігі бар екі жүйе гомоморфты деп аталады.

Модельдеуге қатысты келесі жүйелердің изоморфтылығы мен гомоморфтылығы туралы айтуға болады:
- модельдер мен модельденуші объект;
- бір объектінің түрлі модельдері.

Құрастырылымды объектілер ретінде әлеуметтік және табиғи объектілерден басқа адамның құруындағы объектілер аталады.

Изоморфты (гомоморфты) модельдер тек құрастырылымды объектілерден тұрады. Кейбір модельдердің объектіге изоморфты модель болатындығы туралы тұжырымдар объектіні үйренуден ақпаратты жоғалтпай модельді оқып үйренуге өтуге, модель бойынша объектіні бірмәнді қалпына келтіруге мүмкіндік береді.
Бір объектінін. бір-біріне изоморфты екі түрлі моделі объект туралы бірдей ақпарат береді.

Модель - ғылыми танымның маңызды құралы. Құрал ретінде модель белгіленуі бойынша қолданылуы тиіс.

Кез-келген құралдың шектелген қолданылу аясы бар. Модельдердің сандық, сапалық сипаттамалары:

- моделін оқып үйрену негізінде жасалған модельдеу объектісінің

күйі бағасын дәл болжауға;

- модельдеу мақсатына сәйкес берілген модельдің қолданылу шегін анықтауға қажет.

Құрылған модельдерді:

- модельдің сыртқы түрін түпнұсқаға сай көрнекі құру;

-модельденуші объекті құрылымын толықтай бейнелеу ;

-модельденуші объект күйі туралы көбірек болжамдар жасауға мүмкіндік алу арқылы жетілдіруге болады.

Бұл жетілдірулер модельдеу мақсаты тұрғысынан өзін-өзі ақтауы тиіс. Құрылым - элементтер жиыны мен олардың арасындағы байланыс.
Модельденуші объект құрылымын толықтай бейнелеуді жетілдіру қарастырылатын элементтер санын ұлғайту, олардың арасындағы қатынастар мен қатынастардың параметрлерін нақтылаумен сәйкестенеді.
Ак:параттық модельдердің негізгі сандық бағаларының бірі оның күрделілігі.

Құрылымның күрделілігін оның ең кіші сипаттамасының ұзындығы ретінде түсіну керек (А.Н.Колмогоров бойынша күрделілік).

Алгоритмнің күрделілігі оныІ орындауға жұмсалатын уақыт пен қажетті ресурстар (ЭЕМ, оның жады көлемі, қажетті аппараттық-бағдарламалық жабдықтар) арқылы анықталады.

Құрастырылымды емес объектінің негізгі күрделілік бағасы оның шексіз көп элементтерінің болуымен байланысты. Элементтердің мұндай жиыны дискретті әрі үзіліссіз ұйымдастырылуы мүмкін.

Құрастырылымды емес объектілер негізінен сапалық жағынан бағаланады.

Егер объект күйі белгілі заңдылықтарға бағынып, бастапқы шарттармен бірмәнді анықталса, сәйкес детерминациялық модельдер белгілі физикалық, математикалық, экономикалық заңдар негізінде оның болжамдылығы тұрғысынан сандық бағалануы мүмкін.

Детерминациялық модельдер ортасынан күйі модельденуші объект күйі сияқты бастапқы шарттардың өзгеруіне сәйкес орнықты модельдер бөлінеді.

Модельденуші объектіге түрлі кездейсоқ әсерлердің ықпалын ескеріп, объект күйінің Ықтимал (стохастикалық индетерминациялық) моделін құру қажет. Ықтимал модельдің сандық бағасын ықтималдық теориясы мен математикалық статистика негізінде алуға болады.

Индетерминациялық модельдер орта мән (математикалық күтім), орта мәннің орташа ауытқуы (дисперсия) соңғы көрсеткіштермен сипатталады.

Модельдерді келесі параметрлер бойынша сандық бағалауға болады: Объектінің сырmқы түрін модельдеуде:

- физика-химиялық сипаттамалардан (өлшемі, салмағы, түсі т.с.с.)

берілетін дәлдік (өлшеу қателігі);

- Пропорцияны, масштабты сақтау;

Объект құрылымын модельдеуде:

нақты көрсеткіштер:

- бейнеленетін элементтер мен олардың өзара байланыстарының үлесі (пайыз);

- элементтер салмағы мен олардың арасындағы байланысты бейнелеу

дәлдігі (дөңгелектеу қателігі);

- объект құрылымын деталдау (ірілендіру);

Ықтимал көрсеткіштер:

- элементтер санының орташа мәні мен бұл мәннен орташа ауытқуы

(дисперсия);

- орта бағалардың дәлдігі (сенімділік аралығы);

Объект күйін модельдеуде:

нақты көрсеткіштер:

- объект қатысатын себеп-салдарлық байланыстарды ескеру дәлдігі

(есептеу қателігі);

- дискретті модельдер (дербес жағдайда сандық) кемелмен үзіліссіз процестерді· модельдеуде дискреттеу қадамдары (кванттық уақыт периоды);

- модельдеу процесінің уақыт параметрі бойынша бейнеленуінің пропорционалдылығы (теңөлшемділігі);

Ықтимал көрсеткіштер:

- модельденуші объект күйі параметрлері таратылымының ықтимал зандары;

- объектінің бақыланатын күйі мен оның моделі арасындағы айырымның статистикалық мәнділік деңгейі.

Модельдерді келесі параметрлер бойынша сапалық бағалауға болады:

- модель мен объектінің ұқсастық алмасу дәрежесі (жоғары, орта, төмен дәрежесі);

- модель бойынша объектіні тану дәрежесі (танылды, тануға болады, танылмайды);

- модель бойыша объект күйін алдын-ала болжау дәрежесі.

Модельдеу үшін модель жасау және зерттеу қажет. Кейбір математикалық модельдер есептегіш техниканың көмегінсіз жүзеге аса алады:

ЭВМ-де модельдеу төмендегі сатыларды орындауды қажет етеді[36]:

1.модельдеу мақсатын тұжырымдау;
2.концептуальды модельді жасау;
3.бастапқы мәліметтерді дайындау;
4.математикалық модельді құрастыру;
5.модельдеу әдісін таңдау;
6.модельдеу құралдарын таңдау;
7.бағдарламалық модель жасау;
8.модельдің сәйкестігін тексеру және түзету;
9.тәжірибені жоспарлау;
10ЭВМ көмегімен модельдеу, модельдеу нәтижесін талдау.

Берліген бір S_0 жүйесі үшін $\{S_m\}$ модельдер жиынын құрастыруға болады.

Жүйенің тиімділігін жаңа пайда болғаны жүйе анықтауы керек. Егер жүйенің бірнеше нұсқасы болатын белгілі болса, онда ең жақсы нұсқаны таңдап алуға болады. Жүйенің тиімділігі шыққан шығын мен алған сипаттамалармен салыстырылады:

$E = E(Y_0)$

E -тиімділік көрсеткіші

Y_0-сипаттамалар жиыны

Жүйенің тиімділігін бір критерийлі бағалау сапаның бір дербес көрсеткіші бойынша жүргізіледі [14;23;27;28]. Басқа сипаттамаларға олардың рұқсат етілген өзгерістеріне шек қойылады:

$E = y_{opt}$

$y_{i_{\min}} \le y_i \le y_{i_{\max}}, \quad i = 1,2,...,n$

мұндағы $y_{i_{\min}}, y_{i_{\max}}$ - і-інші сапаны дербес көрсеткішінің төменгі және жоғары шектері, n-ескерілетін сипаттамалардың саны.

Көп критерийлі бағалау нормальданған аддитивті критерий қолданылады:

$$E = \sum_{i=1}^{n} b_i \gamma(y_i)$$

$\gamma(y_i)$ функциясы і-інші сипаттаманың өлшемін жою және $\gamma(y_i) \in [0,1]$ шартын орындалуына сай алынады.

Салмақ коэффициенттері $\sum_{i=1}^{n} b_i = 1; \quad b_i > 0$ шартын қанағаттандырады.

Жеке жағдайда $0 \le y_i \le y_{i_{\max}}$, $\gamma(y_i) = y_i / y_{i_{\max}}$ болғанда нормальданған аддитивті критерий сызықтық формаға енеді.

Модельді құрастыру барысында сипаттаудың төменгі баспалдақтары: концептуальды, математикалық, бағдарламалық сатылары болып өтіледі.

Концептуальды модель жасау кезеңдері [37].

1.Бағытты тағайындау. Концептуальды (мазмұнды) модель сөз жүзінде жүйенің құрамы мен құрылымын, компоненттердің қасиеттерін олардың арасындағы себеп-салдардың байланыстарды анықтайды. Мұнда зерттелетін жүйенің табиғаты мен параметрлері туралы мәліметтер, олардың арасындағы байланыс түрлері мен дәрежесі, жүйенің жұмысында жалпы үрдістегі әрбір құбылыстың орны мен маңызын қарастырады.

2.Стратификация. Стратификация модельді егжей-тегжейлендіру деңгейін таңдау. Егжей-тегжейлендіру деңгейлері кейде стратификация дейді. Егжей-тегжейлендіру деңгейі модельдер кезінде өзгертуге болатын параметрлермен жиі анықталады. Бұндай параметрлер қажетті сипаттамаларды анықтауды қамтамасыз етеді. Басқа параметрлер мүмкіндігінше модельде ескерілмеуі керек.

1.Детализация – мұнда шығыс параметрлерінің кіріс параметрлеріне тәуелділігі белгілі не алынуы мүмкін элементар операция ұғымы енгізіледі. Бұл кезде үлкен өлшемді модель алынатын болса, онда оны иерархиялы модельді структураға түрлендіруге болады. Мұндай структураның әрбір модулі төменгі рангалы модульдердің жиынынан, соның ішінде элементар операциялардан тұрады.

2.Структураландыру және басқару жүйенің құрама бөліктерінің арасындағы байланыстар заттық немесе ақпараттық болуы мүмкін.

Заттық байланыстар бір элементтен екінші элементке өндіру өнімінің ауысуын көрсетеді. Ақпараттық байланыстар басқару әсерлерінің және күй туралы ақпараттардың құрама бөліктерінің арасында берілуін қамтамасыз етеді. Қарапайым жүйелерде ақпараттық байланыстар болмауы мүмкін, ол структуралық басқару ұстанымына сәйкес [27].

Күрделі жүйелерде басқаруға қай құрама бөлікке, қандай бастапқы объекті, түрлендіру үшін қандай операцияны орындау, қашан, қайдан алу және қайда беру туралы нұсқаулар кіреді. Мұндай жүйелер басқарудың

бағдарламалық немесе алгоритмдік үрдісіне сәйкес жұмыс істейді. Концептуальды модельде жұмыстық жүктемелі алгоритмді басқару немесе барлық елеулі ережелер нақтылануы қажет.

Локализация. Локализация сатысында сыртқы орта туралы көрініс модельдің құрамына компонент түрінде кіретін сыртқы әсер генераторлары түрінде жүзеге асады. Мұнда жұмыс жүктемесі генераторы, басқару және ұйтқу әсерлер генераторлары кіреді. Шығыс әсерлерін қабылдағыштар, әдетте, модельге енбейді.

Үрдістерде бөліп көрсету. Жүйенің жұмыс істеуі затты, энергияны немесе ақпаратты түрлендіретін технологиялық үрдістерді орындауды қамтиды. Күрделі жүйелер, әдетте, бір мезгілде бірнеше үрдістер жүзеге асады. әрбір үрдіс жеке элементар операциялардың белгілі бір тізбегінен тұрады.

Операциялардың бір бөлігі жүйенің әр түрлі белсенді компоненттерімен параллель орындалуы мүмкін. Технологиялық процесс алгоритмдерді көрінісінің бір түрімен беріледі. Бірнеше үрдістерді параллель орындалуын қамтамасыз ететін бағдарламалық жүйелі жүйелерде, параллель жұмыс жасайтын үрдістер жиынымен жұмыс жасайтын алгоритмдер болады.

Концептуальды модель S параметрлер және X сыртқы әсерлер жиынын анықтайды. Модельдеу кезінде бастапқы мәлімет түрінде пайдаланатын сандық параметрлердің нақты мәнін анықтау қажет.

Мәліметтерді жинау үшін төмендегілерді ескеру қажет:

- параметрлердің мәндер детерминделген немесе стохастикалық болуы мүмкін;
- барлық параметрлер стационар емес;
- жоқ жүйе туралы сөз болуы мүмкін (жобалауда, түрлендіруде).

Параметрлердің бір қатарлы табиғаты жағынан кездейсоқ шамалар болып келеді. Модельдің қарапайымдандыру мақсатында олардың бір қатары детермендірілген орташа мәндер түрінде алынады. Бұл жағдай шамаларды ауытқуы үлкен болмауы шарт немесе модельдеу мақсатына орташа мәндерді қолдану арқылы жетуге болады.

Кей жағдайда детерминген параметрлер кездейсоқ шамалар арқылы өрнектеледі. Мысалы, бағдарламаны көп қайталап орындағанда өңделетін сәндер шамасы өзгереді [38]. Мәліметтердің нұсқаларының жиынын ықтималдық үлестіруін, берілген заңымен берілген кездейсоқ шамалар түрінде қарастыруға болады.

Кездейсоқ шамаларды теориялық үлестіру заңы түрінде көрсетуге болады. Үлестіру заңы түрін таңдай алу төмендегідей болады.

Параметрлердің шамалары жиыны бойынша салыстырмалы жиіліктер гистограммасы тұрғызылады. Гистограмма әр түрлі теориялық үлестіру заңдарының тығыздық қисықтарымен салыстырылатын қисықпен апроксимиляцияланады. Сәйкес келуінің ең жақсысы бойынша заңдардың

бірі таңдалады. әрі қарай тәжірибелік шамалар бойынша осы үлестірудің параметрлері есептеледі. Тәжірибелік және теориялық үлестірулердің сәйкестігін тексеру сәйкес критерйлерінің бірімен жүзеге асады: Пирсон (хи-квадрат [14,18,28,39], Колмогоров [6,11,18,20,40], Смирнов [6,16,25,40,41], Фишер [16,23],

Сыртқы әсерлерге тән, уақытқа тәуелді кездейсоқ параметрлер бойынша мәліметтер жинау аса қиындық туғызады. Параметрлердің стационар еместігін ескермеу модельдің сәйкестігіне елеулі әсер етеді. Жүйенің әрбәр құрамдас бөлігінде сыртқы әсер параметрлері мен шығыс сипаттамаларының арасында функциональды байланыс болады. Кейде функциональдық байланыс жеңіл табылады. Алайда кейбір құрама бөліктер үшін кіріс параметрлері мен шығыс сипаттамалары үшін тек тәжірибелік мәліметтер алуға болады.

Бұл жағдайда функциональдық тәуелділік туралы гипотеза, яғни оны белгілі бір математикалық теңдеулер бойынша аппромиксация енгізіледі. Жиналған мәліметтер бойынша екі немесе көп айнымалылар арасындағы тәуелділікті өңдеу регрессиялық, коррелялық немесе дисперсиялық талдау арқылы орындалуы мүмкін.

Математикалық теңдеудің түрін зерттеуші таңдайды. Содан кейін регрессиялық анализ арқылы теңдеудің коэффициенттері есептеледі. Қисық пен тәжірибелік мәліметтерді жуықтау ең кіші квадраттар критерийі бойынша бағаланады.

Таңдап алған тәуелділік тәжірибелік мәліметтермен қанша дәл екендігін анықтау үшін корреляциялық анализ қолданылады.

Жаңа жүйені жобалау үшін фактылар жинау мүмкіндік болмайды. Мұндай параметрлер үшін олардың мүмкін мәндері туралы гипотеза тұжырымдалады.

Мәліметтерді жинау және өңдеу кезеңі оларды сыртқы және ішкі, тұрақты және айнымалы, үздіксіз және дискретті, сызықтық және сызықтық емес, детерминизацияланған және стохастикалық деп классификациялаумен бітеді. Парметрлердің айнымалы саны үшін олардың өзгеру шщекарасын, ал дискретті шамалар – мүмкін мәндері анықталады.

Математикалық модель – техникалық объектінің физикалық қасиеттерін сәйкес бейнелейтін математикалық объектілер мен олардың өзара қатынастарының жиыны [5,16,23].

Күрделі техникалық жүйенің жобалаудың әр түрлі этаптары мен кезеңдерінде әр түрлі математикалық модельдер қолданылады. Математикалық модельдер дифференциалдық теңдеу, алгебралық теңдеулер жүйесі, қарапайым математикалық өрнектер, бинарлық қатынастар, матрица және т.б. түрінде болуы мүмкін. Математикалық модельдің теңдеулері физикалық шамаларды байланыстырады.

Математикалық модельдерге сәйкестілік, үнемділік, әмбебаптық талаптары қойылады [16,43]. Модель зерттелетін қасиеттерді барынша дәл бейнелейтін болса, сейкес деп есептеледі.

Техникалық объектілердің жобалау кезінде пайдаланатын математикалық модельдері объектілердің жұмыс жасау үрдісін және шығыс параметрлерін бағалауға арналған. Олар жобалаудың нақты есебін шешуге маңызды объектінің физикалық қасиеттерін бейнелеуі керек. Математикалық модель барынша қарапайым болуы керек, сонымен қатар талданатын үрдіске сәйкес сипат беруін қамтамасыз ету керек.

Математикалық модельдердің келесі түрлерін қолданады: детерминделген және ықтималдық, теориялық және тәжірибелік факторлық, сызықтық және сызықтық емес, динамикалық және статикалық, үздіксіз және дискретті, функциональды және структуралық.

Математикалық модельдерді формасы бойынша төмендегідей топтарға бөледі:

1.Инвариантты модель – (дифференциалдық және алгебралық) теңдеулер жүйесіне құрылған математикалық модель, осы теңдеулерді шешу әдістерінен тыс.

2Алгебралық модель – модельдердің арақатынасы таңдап алған санау әдісімен байланысты және алгоритм түрінде (есептеу ретінде) жазылған.

3.Аналитикалық модель - ізделетін айнымалылар берілген шамалардан ашық түрде тәуелді. Мұндай модельдерді физикалық заңдар негізінде немесе кестелік интегралдарды пайдалана отырып, алғашқы дифференциалдық теңдеулерді тікелей интегралдау арқылы алады. Оларға тәжірибе нәтижесінің нәтижесінде алынған регрессиялық модельдер жатады.

4.Графикалық модель – графиктер, сәйкесті схемалар, динамикалық модельдер, диаграммалар және сол сияқтылар. Графикалық модельдерді пайдалануда графикалық элементтердің шартты белгіленулерімен инвариантық математикалық модельдің компоненттерінің бір мәнді сәйкестік негізінде алынған репрессиялық модельдер жатады.

Математикалық модельдер шығыс, ішкі және сыртқы параметрлер арасындағы функциональдық байланыс түрінде де бола алады. Математикалық модельдерді функциональдық және структуралық деп бөлу техникалық объектінің бейнеленетін қасиеттерінің сипатымен анықталады.

Структуралық модельдер объектінің структурасын бейнелейді және структуралық талдау есептерін шешу кезінде пайдаланылады. Структуралық модельдердің параметрлері болып техникалық объектілердің функциональдық немесе конструктивтік элементтердің белгілері алынады, олар арқылы объектінің бір нұсқасы екіншісінен ажыратылады. Мұндай модельдер кестелер, матрицалар және графиктер

түрінде болып келеді. Олар техникалық объектіні таңдауды мета деңгейінде кеңінен пайдаланылады.

Енді формальды жүйе ретіндегі комбинаторлы логиканың сипаттамасына өтейік. Математика практикасына сәйкес, теорияның келесі элементтерін анықтау қажет:

- Әліпби (алфавит)
- Тұжырымдау
- Аксиалалар
- Қорытынды ережесі.

Мынаны еске салған жөн әліпби (алфавит) ретінде қандай да бір формализацияны шартты түрде белгілеуге жарамды жағдайдағы
символдардың жиынын түсінеміз.

Тұжырымдау математикалық теорияның терминальды символдарының қалыптасу ережелерін бекітеді.

Аксиома деп – ақиқаттылығын дәлелдеуді қажет етпей-ақ ақиқат деп есептеуге болатын қарапайым тұжырымдарды айтамыз.

Қорытынды-ережесі теорияда зерттелетін бір символдардың (обьектілердің)
өзге бір обьектіге айналуының ережесін анықтайды. Комбинаторлы логиканың әлбиін (алфавитін) қарастырайық.

Мұнда төрт түрдегі элементтерге жол тарайды.

1. Константтар
2. Айнымалылар
3. Комбинаторлы өрнектер (немесе (или) басқаша (иначе) термдер).
4. Арнайы символдар

Константтар C1,C2, ... латын алфавитімен сипатталады.

Екі комбинаторлық термнің конверттелуші – бірі комбинаторлық термнің өзге комбинаторлық термге айналатындығын білдіреді. Конверттелушілік қатнастары коптеген қатнастардағы қайта белгілеулерді де модельдейді, яғни бұдан бұрын айтылғандай программалау процесін білдіреді.

Келесі акциомалар конверттелушілік қатынастарының қасиеттерін білдіреді.

$$Ix=x;\ (k)Kxy=x;\ (S)Sxyz=xz(yz). \quad (I)$$

(I)-бұл аксиомасы тепе- теңдік комбинаторының (функциясының) бар екендігін білдіреді, яғни дәлірек айтқанда кез-келген аргументтің өзінде бейнеленетін, тепе-тең өзгерістердің бар екендігін білдіреді.

Аксиомасы бірінші проекцияны алу комбинаторының (функциясының), яғни реттелген қосақтың алғашқы элементінің немесе тізімнің бірінші элементтінің бар екендігін білдіреді. Бұл акциоманың тізімдерді амалдаушы функционалдық программалау тілене жақын екендігі айқындалып отыр және ол тізімнің алғашқы элементін алудың фундоментальды операциясын сәйкес келеді.

Программалауға қатысты функцияналдық амал. Функционалдық амалды негіздейтін теориялық жұмыстардың пайда болуы XX ғасырлың 20-шы, 30-шы жылдарына жатады. Кейінірек бір көз жеткізетініміздей, пограммалау теориясы программалау практикасынан едәуір түрде озып жүреді. Амалдың математикалық негізін қалыптастырған маңызды жұмыстар осы теорияны жүзеге асыруға мүмкіндіктері бар компьютерлер мен программалау тілдері пайда болғаннан көп уақыт бұрын жазылған болатын.

Алғашқы жүзеге асыруға келсек, ол XX ғасырдың 50-шы жылдарында LISP тілінің формуласында пайда болды енді әнгіме осы жайлы болмақ.

Біріншіден мынаны еске сала кетуіміз қажет; функционалдық маңызды сипаттамасы ол функционалдық программалау тілінде құрылған кез келген программа, аргументтері функция болып табылуы мүмкін функция ретінде қарастырылуы мүмкін.

Функционалдық амалдан бұрын айтылғандай шығу тегі LISP программалау тіліне барып тірелетін бір топ тілдер дүниеге келді. Кейінрек 70-ші жылдары ML тілінің бастапқы нұсқасы құрылған болатын. Ал кейінрек бұл тіл әсіреесе SML тіліне сондай-ақ бір қатар өзге тілдерге қарай дамыды, 90-шы жылдары құрылған Haskell тілі болып табылды.

Функционалды программалау тілдерін жүзеге асырудың маңызды артықшылығы мәліметтерді сақтау үшін компьютер жадын автоматтандырылған динамикалық күйде бөліп тарту болып табылады. Бұл жағдайда программист мәліметтерді бақылау қажеттілігінен арылады, ал егер ал қажет болған жағдайда ол (программист) жадына программаға қажеті жоқ мәліметтерден тазалайтын “қоқысты жинау” функциясын іске қоса алады.

Күрделі программалау функционалды амал кезінде функционалды агрегаттау арқылы құрылады. Бұл жерде программа мәтіні функцияны білдіреді, ал оның кейбір аргументтерін функция ретінде қарастыруға болады. Міне осылайша кодты қайта пайдалану, құрылымы илтеративті тілдің процедурасына қарғанда математикалық жағынан мөлдір болып келетін бұдан бұрын сипатталған функцияны шақырып алуға барып тіреледі.

Функцияның, функционолдың программалау тілдері үшін табиғи формализім болып табылатындығынан, функцияларымен байланысты болып келетін прграммалаудың алуан түрлі аспектілерінің жүзеге асыру елеулі түрде ықшамдалады. Көкейкөзді түрде ашық (мөлдір) болатын ол рекурсивті фуекция, яғни аргумент ретінде өздерін-өздері шақыратын функциялар. Мәліметтердің рекурсипті құрлымдарын өндеуді жүзеге асыруда табиғи бола бастайды.

Конверсия ретінде қисаптау обьектілердің (программалауда – функциялармен мәлеметтерді) бір формадан екінші формаға уыстыруды түсінеміз. Математикадағы алғашқы есеп ол өрнек

формаларын ықшамдауға деген ұмтылыс болды. Программалауда дәл осы есеп айтарлықтай мәнді болмайды. Әйтседе бұдан былай байқайтынымыздай алғашқы формалдау ретіндегі ламбда – қисаптарды пайдалану программа түрін ықшамдауға жағдай жасауы мүмкін, яғни программалық кодтың тиімділенуіне алып келу мүмкін.
Алгоритмдік тілдердің осы деңгейлерін сипаттайық:

Сұрау алу тілдері (процедуралық еместілдер) қолданбалы программалардың кейбір топтарымен диалогты жүзеге асыру үшін арналған, бұлар ұқсататын модельдеу тілдері, дәлірек айтқанда SLAM тілі және тағы басқа;

Жоғары деңгейдегі тілдер (проблемалық – бағытталған тілдер) белгілі бір, бірақ барынша кең көлемдегі мәселелерді шешуге, мәселен: есептеу сипатындағы немесе мәтіндерді өңдеу секілді мәселелерді шешуге арналған,
мысалға айтар болсақ олар FORTRAN, BASIC, LISP және тағы басқа тілдер;

Ассемблер (тілдер тобы) машиналық командаларды үлкейтуге және символдық (мнемоникалық) жазба үшін арналған;

Микрооперациялар тілдері (микропрограммаларды жасау тілдері) – бұл машиналық операциялардың нағыз өзі.

Тілдерді пайдалану тілдері мен қолдану салалары бойынша шартты түрде мынандай типтерге бөлуге болады (бұл олардың толық топтамасы емес).

1. Процедуралық тілдер.
2. Процедуралық емес тілдер.
3. Функционалды программалау тілдері.
4. Модельдеу тілдері.
5. Талдау өзгерістер тілдері.
6. Эвристикалық тілдер.
7. Сипаттау тілдері, обьектілік тілдерді көрсету немесе метатілдер.

Функциональдық модельде техникалық объектілердің жұмыс істеу үрдісін сипаттайды, теңдеулер жүйесі түрінде болып келеді. Олардың иерархиялық деңгейде, функионалдық, конструкциялық және технологиялық жобалау этаптары мен стадияларында кең пайдаланады.
Функциональдық модельдері алу әдістері бойынша бөледі:

- Теориялық модельдер – объектінің жұмыс істеуінің физика үрдістерін сипаттау негізінде алады.

- Тәжірибелік модельдер – объектіні "қара жәшік" түрінде қарастыра отырып, оның сыртқы ортадағы сипаты негізінде алады.

Теориялық модельдерді жасау кезінде физикалық және формальды қатынастар алынады. Физикалық қатынас объектіге тікелей физикалық

заңдарды қолдануға әкеледі. Формальды қатынас теориялық та, тәжірибелік те модельдерді құру кезінде қолданылады.

Функциональды математикалық модельдер:

1.Сызықтық модельдер тек фазалық айнымалылар мен олардың туындыларынан тұратын сызықтық функциялар

2.Сызықтық емес математикалық модельдер өз құрамында фазалық айнымалылар мен олардың туындыларының сызықтық емес функцияларын қамтуы мүмкін [16,43,45].

Егер модельдеу кезінде техникалық объектінің инерциялық қасиеттері және (немесе) объект не сыртқы ортаның параметрлерінің уақыт ішінде өзгеріс ескерілетін болса, ондай модель динамикалық деп аталады. Кері жағдайда модель статикалық болады. Динамикалық немесе статикалық модельді таңдау техникалық объектінің жұмысымен анықталады. Динамикалық модельді математикалық түрде бейнелеу жалпы жағдайда дифференциалдық теңдеулер жүйесімен іске асырылуы мүмкін, статикалық – алгебралық теңдеулермен беріледі. Динамикалық модель сол сияқты интегралдық теңдеулер, қосымша функциялар түрінде де, ал аналитикалық формада – фазалық координаталар немесе техникалық объектінің шығыс параметрлерінің тікелей уақытқа тәуелді түрінде көрсетіледі.

Жүйенің жұмыс істеуі технологиялық үрдісті орындауында болып табылады. Бұл жағдайда жүйеде бір мезгілде бірнеше технологиялық үрдіс жүзеге асырылады. Технологиялық үрдіс белгілі бір операция тізбекті құрайды. Операциялардың бір бөлігі жүйенің белсенді ресурстарымен параллель орындалуы мүмкін. Технологиялық үрдіс алгоритмдердің бір әдісімен беріледі.

Бірнеше технологиялық үрдісті параллель орындауды қамтамасыз ететін бағдарламалық басқару принципі бар жүйелерде үрдістердің тобымен басқару алгоритмдері бар. Бұл алгоритмдердің негізгі мақсаты – екі немесе одан да көп үрдістер бір ресурсты қажет ектенде пайда болатын қақтығыстық жағдайларды шешуге болады. Басқару алгоритмдерінің A_0 жиыны кіріс әсерлерінің X_0 параметрлері және S_0 құрама бөліктермен қоса жүйенің жұмыс істеу динамикасын жасайды. Көбіне басқару алгоритмдері $A_м$ модельдеу ыңғайлы түрде беріледі. Берілген жүйенің жұмысының динамикасын сипаттау имитациялық модельдеуде ыңғайлы болады, берілген жүйенің сипаттамаларын анықтаудың табиғи жолы болып табылады:

$$Y = F(X, S, A, T),$$

мұнда Ғ – шығыс сипаттамаларын анықтайтын операциялар жиыны, Т – модельдеудің календарлық уақыты 8.

Басқарудың структуралық принципті жүйелер үшін жүйенің жұмыс істеу динамикасын сипаттау басқаша қаралады. Әрбір құрама бөлік үшін белгілі бір S параметр немесе бірнеше параметр таңдап алынады. Олардың

мәндері жұмыс істеу барысында өзгеріп тұрады және Z (t) алынған уақыт сәтінде жүйенің күйін көрсетеді. Жүйенің барлық n=1, элементі бойынша параметрлердің жиыны (Z_n) жүйенің Z (t) күйін көрсетеді. Жүйенің жұмыс істеуі күйлердің ауысуының тізбегі ретінде Z (t_0), Z (t_1), ... Z (T) беріледі. Жүйенің бола алатын күйлерінің жиыны (Z) күй кеңістігі деп аталады. Жүйенің ағымдағы күйі t уақыт сәтінде ($t_0<t \leq T$) n өлшемді күй кеңістігіне нүкте түрінде, ал жүйенің үрдісінің T уақыт ішінде жүзеге асыуы белгілі бір түрде бейнеленеді.

Жүйенің берілген бастапқы күйі $Z^0=Z(t_0)$ бойынша Z(t)=H(X, Z^0, Z, t) тәуелділіг белгілі болса, оның күйін (t_0T) интервалындағы кез келген t сәтінде анықтауға болады. Бұл жағдай шығыс сипаттамалары У=G(ZT) өрнегімен анықталады.

Ғ және G операторлары шығыс операторлары, ал Н операторы өту операторы деп аталады. Жалпы модельді конструктивтік модельге айналдыру мақсатында айнымалы мен операторлар жиынның қасиеттерін нақтылау керек. Жүйелердің белгілі бір кластары үшін формальді және математикалық әдістер жасалған. Олар жүйенің жұмыс істеуін сипаттай алады, ал кейбір жағдайда - аналитикалық зерттеу жүргізе алады.

Формальданған схемалардың жалпы басым бөлігі агрегативтік жүйе түрінде сипаттау болып табылады [34, 40, 43, 44, 46]. Бұл әдіс кіріс және шығыс әсерлерін «сигналдар» тобына кіретін «хабар» түрінде берілетін жүйелерді сипаттауға барынша икемделген әдістің негізіне жүйенің құрама бөлігі ретінде агрегат ұғымы алынады. Агрегаттың математикалық моделі нақтыланған кіріс әсерлерін, күйлерді және өту және шығыс операторының тәуелсіздігі түрінде өрнектеледі. Зерттелетін объектілерді агрегаттық жүйелер түрінде математикалық сипаттау имитациялық модельдің әмбебап әдістерін қолдануға мүмкіндік береді.

Көптеген дискретті күйлерден, уақыт ішінде үздіксіз жұмыс жасайтын шығыс және кіріс әсерлерінен тұратын стохастикалық жүйелерді сипаттау үшін стохастикалық торлар қолданылады. Стохастикалық тор жалпы қызмет ететін бір жүйеден екінші жүйеге өтетін сұраныстардан тұратын жүйелер жиынынан тұрады.

Егер модельде кездейсоқ факторлар әсері ескерілмесе, өту және шығыс операторы үздіксіз болса, онда тәуелдіктер дифференциалдық теңдеулер түрінде беріле алады:

$$\frac{dz}{dt} = h(z(t), x(t), t),$$

$$y = g(z(t), t),$$

Мұнда h, d – күйлерді шығыстарына сәйкесті функциялар векторы;

x, z, y – кіріс әсерлерінің, күйлердің, сәйкесті шығыс әсерлерінің векторы.

Күйлері $t_0, t_1, t_2, \ldots,$ дискретті уақыт моменттерінде жүйелер автоматтар деп аталады. Егер уақыт бірлігіне $\Delta t = t_i - t_{i-1},$ такт алынған болса, онда

дискретті уақыт моменттері 0, 1, 2, ... түрінде алынады. Әрбір дискретті уақыт моментінде t_0-ден өзгеше, автоматқа кіріс сигналы x(t) түседі. Оның әсерінен автомат өту функциясына
$z(t) = \varphi_a(z(t-1), x(t))$

Жаңа күйге көшеді, шығыстар функциясымен анықталатын
$y(t) = \psi_a(z(t-1), x(t))$. шығыс сигналын береді.
Егер автомат шектелген Z күй , X кіріс сигналдар және У шығыс сигналдарын жиынымен сипаттаса, автомат шектелген автомат деп аталады. Шектелген автоматтарды өту және шығыс функциялары кестелер, матрица және графиктермен беріледі.
Дискретті уақытта жұмыс істейтін стохастикалық жүйелерді ықтимал автоматтармен бейнелеуге болады. Ықтимал автоматтар өту функциясын бір нақты күй анықтамайды, күйлердің жиынында ықтималдық үлестірумен анықталады, ал шығыс функциясы шығыс сигналдардың жиынындағы ықтималдық үлестірумен анықталады. Ықтимал автоматтардың жұмыс істеуі Марковтың тізбектер аппаратының көмегімен зерттеледі. Автоматтар түрінде берілетін жүйелерді талдау үшін аналитикалық және имитациялық әдістер қолданыла алады.
Параллель өзара әсерлесетін үрдістер жиынымен жұмыс жасайтын жүйелерді зерттеу үшін қолданылатын автоматтар класында Петри торы [6, 16, 24] және оның әр түрлі модификациясы, Керк торы, Е-тор кеңінен қолданылады.

Математикалық модель әр түрлі әдістермен – аналитикалық немесе имитациялық әдістермен зерттеледі. Кейбір жағдайларда имитациялық модельдің болуы оңтайландырудың математикалық әдістерін қолдануға мүмкіндік береді. Аналитикалық әдістерді қолдану үшін математикалық модельді сыртқы әсерлермен жүйенің параметрлері мен сипаттамаларының арасындағы тәуелділікті ашық аналитикалық тәуелділікке түрлендіру керек. Бірақ бұл салыстырмалы қарапайым жүйелер үшін және зерттелетін объект үшін жақсы жасалған теория болған жағдайда сәтті болады.

Имитациялық модельдеу негізінде жүйенің оригиналдың динамикалық үрдістері және операциялардың ұзақтығы мен уақыт реті қатынасы сақтала отырып, абстракциялық модель имитацияланады. Сондықтан имитациялық модельдеу әдісін алгоритмдік немесе операциялық деп атауға болады.
Имитациялық модельдеу әдіс зерттелетін жүйелердің класына, модельдің уақытына өтуіне, жүйенің және сыртқы әсерлердің айнымалыларының параметрлерінің санына байланысты бөлінеді. Модельдеу әдістері дискретті, үздіксіз және дсикретті-үздіксіз араласқан жүйелер үшін жасалынады [47].

Модельдің уақытын өту түріне байланысты модельдеу әдісі уақыт интервалының өсімшесімен әдісі және уақыттың ерекше күйге дейін өтуімен бөлінеді. Бірінші жағдайда модельдік уақыт белгілі бір Δt шамасына өзгереді. Әрі қарай эүйенің элементтерінің күйлері мен шығыс әсерлерінің осы уақыт ішіндегі өзгерістері анықталады. Осыдан кейін модельдік уақыт тағы да Δt шамасына өзгертіліп, жұмыс қайталанады. Осылай T_m модельдеу периодының соңына дейін жалғаса береді. Бұл модельдеу әдісін «Δt» принципі деп атайды.

Екінші жағдайда модельдік уақыттың ағымындағы t моментінде $t_1 > t$ болу уақыты анықталған, болашақ ерекше күйлер – дискретті кіріс әсерінің түсуі, қызмет көрсетудің аяқталуы және т.б. қолданылады. Ең бірінші болатын ерекше жағдайда таңдап алынады да, модельдік уақыт осы күй болғанша жылжытылады. Содан кейін жүйенің таңдалған ерекше күйге реакциясы талданады. Талдау кезінде жаңа ерекше күйдің болуының сәті анықталады. Сосын келесі ерекше жағдайлар талданып модельдік уақыт келесі жақын мәніне жылжытылады. Осылай T_m модельдеу периоды соңына дейін жалғаса береді. Әдісті тек болашақ келесі ерекше күйлер болуы сәтін анықтау мүмкін болғанда ғана қолдануға болады.

Жүйенің параметрлері және сыртқы әсерлер детерминизацияланған немесе кездейсоқ болуы мүмкін. Осы белгі арқылы детерминизацияланған және статистикалық модельдуді айырады. Статистикалық модельдеу кезінде сынақтар әдісі немесе Монте-Карло әдісі алынған [18,24,32].

Математикалық модельді құрастыру кезінде алынған формализация жүйесін де айырады – алгоритмдік (бағдарламалық) немесе структуралық (агрегаттық) түрлері. Бірінші жағдайда үрдістер жүйенің компоненттерін (ресурстарын) басқарады, екінші жағдайда – компоненттер үрдістерді басқарады, жүйенің жұмыс істеу тәртібін анықтайды.

Қазіргі ЭВМ күрделі тармақталған динамикалық жүйелерді модельдеуге мүмкіндік береді. Тармақталу факторы маңызды орын алады және локалдық есептегіш желілер негізінде көп процессорлы санау жүйелерін құруды қарастырады. Сондықтан осы жүйелерді модельдеу үшін тармақталған көп процессорлық санау жүйелерін қолданудың болашағы зор.

Жалпы мақсаттағы алгоритмдік тілдер модельдеудің аналитикалық әдістеріне қолдануға болады. Имитациялық алгоритмдерді бағдарламаларын жасауда жүйелердің тәртібінің алгоритмдері параллель болып келеді, яғни әр уақыт сәтінде бірден ортақ түрлендіруді қолдануды қажет етеді. Жалпы тілдерді пайдалану кезінде параллель үрдістерді бағдарламалық имитациялау параллель үрдістердің псевдопараллель дамуын ұйымдастыруға әкеліп тірейді, ал бұл бағдарламалау үшін жеткілікті түрде күрделі.

Екінші қиындық имитациялық алгоритмдердің мәліметтр көлемін алдын ала бағалау қиын. Сондықтан жадының динамикалық таралуын

пайдалануды талап етеді, оны бағдарламалау тілі әрқашан қамтамасыз ете алмайды.

Имитациялық модельдеу бағдарламасын құрастыру кезінде модельдеудің үлкен класына ортақ мәселелер туындайды:

- алгоритмдерді псевдопараллель ұйымдастыру;
- жадының динамикалық таралуы;
- түпнұсқаның жұмыс істеуінің астрономиялық уақытына сәйкес модельдік уақыт пен операциялар;
- кездейсоқ үрдістерді имитациялау;
- оқиғалардың массивні жүргізу;
- модельдеудің нәтижелерін жинақтау және өңдеу.

Жоғарыда аталған есептерді шешу модельдеу тілдерінің құралдарымен толықтай немесе жартылай шешеді.

Бағдарламалау ережелері мен стуктурасы бойынша модельдеу тілдері жоғары деңгейлі процедуралық бағытталған алгоритмдік тілдерге ұқсас. Модельдеу тілдерінің операторлары күрделірек процедураларды орындайды, сондықтан олардың деңгейі жоғары болады және бағдарламаны құрауды жеңілдетеді. Модельдеу тілдері математикалық модель жасауға формализацияланған негіз бола алады.

Бағдарламалау үрдісін әрі қарай және жеделдету оларды автоматтандыруды қажет етеді. Қазіргі кезде имитациялық модельдерді автоматтандырудың бірқатар жүйесі құрылған.

Модельдің сәйкестігі көп себептерден бұзылады; сыртқы шарттар және жұмыс істеу тәртібін идеалдау; кейбір параметрлерді ескермеу. Сыртқы әсерлер жүйенің құрылымы ерекшеліктері туралы толық мәліметтердің болмауы алынған апромиксациялар, интерполяциялар, гипотезалар мен болжаулар да модельдеу нәтижелері мен нақталынатын елеулі өзгеше болады.

Сәйкестіктің қарапайым өлшемі ретінде түпнұсқаның белгілі бір сипаттамасының y_0 модельдің y_m сипаттамасынан ауытқуын алуға болады.

$\Delta y = |y_0 - y_m|$ немесе $\Delta y = |y_0 - y_m| / y_0$

Егер ауытқуы шектік Δy шамасынан рұқсат етілген P_m ықтималдылықтан артық болмаса модель түпнұсқамен сәйкес деп санауға болады.

$$P(\Delta y < \Delta) \geq P_\Delta$$

Іс жүзінде берілген сәйкестік критерийін төмендегі себептермен алу мүмкін болмайды:

- жобаланатын немесе модернизацияланатын жүйелер үшін y_0 сипаттамасының мәні туралы мәлімет болмауы;
- жүйе бір емес, ауытқуының шамасы әр түрлі болатын бірнеше сипаттамалармен бағаланады;
- сипаттамалар кездейсоқ, кейде стационар емес шамалар немесе функциялар болуы мүмкін;

- шектік Δ ауытқулар мен рұқсат етілген P_{Δ} ықтималдылық мәндерін априорлы дәл берілу мүмкіндігінің болмауы.

Осыларға қарамастан сәйкестікті тексеру қажет, әйтпесе модельдеудің қате нәтижелерін қате шешім қабылдауы мүмкін.

Іс жүзінде сәйкестік модельдеу нәтижесінің эксперттік талдау арқылы жүргізіледі. Тексерудің төмендегі түрлерін атап өтуге болады:49

- компоненттер модельдерін тексеру;

-сыртқы әсерлер модельдерін тексеру (апроксимациялар мен гипотезаларды математикалық әдістермен бағалау);

- жүйенің жұмыс істеуінің концептуалды моделін тексеру;

- формализацияланған және математикалық модельді тексеру;

- өлшеу әдістерін және шығыс сипаттамаларын есептеу;

- бағдарламалық модельді тексеру.

Егер сәйкестік тексеру нәтижесінде модель мен жүйенің алшақтығы үлкен болса, модельді түзету немесе калибрлеу қажеттілігі туады. Бұл кезде өзгертулердің келесі типтері бөлініп алынады: глобальді, локальді және параметрлік.

Модельді коррекциялау стратегиясы ең алдымен глобальді өзгертуер енгізуге, содан кейін локальді және параметрлік өзгертулерді енгізуге бағытталуы тиіс. Параметрлік өзгертулер тактикасын жасау үшін модельдің оның параметрлерінің вариациясына сезгіштігін талдау үлкен маңызға ие.

Модельді сәйкестікке тексеру және коррекциялау кезеңі модельдің жарамдылық аймағын анықтау және бекітумен аяқталады. Жарамдылық аймағы деп орындалу кезінде модельдің нәтижесі дәлдігі жеткілікті шекте болатын шарттардың жиынын айтады.

Модельдеудің мақсаты жасалған модельді зерттеу жолымен жүзеге асады. Зерттеу жүйенің белгілі бір модельдің басқарылатын түрлі айнымалы мәндерінде шығыс сипаттамаларын анықтау тәжірибелерін өткізу арқылы жүргізіледі. Тәжірибелерді белгілі бір жоспармен жүргізу керек.

Санау және статистикалық модельдеу тәжірибелерін жоспарлау аса маңызды болады. Бұл басқарылатын параметрлердің мәндерінің мүмкін әсерлесуінің көп болуына байланысты, ол әрбір тәжірибе параметрлердің белгілі бір үйлесуімен жүргізіледі. Мысалы, әрқайсысының үш мәні болатын бес басқарылатын параметрдің алмастыру саны 243, бес мәні он параметрдің алмастыру саны 10 миллионға жуықтайды. Сондықтан параметр алмастыру санын және тәжірибені өткізу ретін таңдау қажет болады. Бұл стратегиялық жоспарлау деп аталады.

Жоспарды құру жүйенің жұмыс істеу сапасын басқаруға керек параметрлер мен сипаттамалар сапасын анықтау негізінде, модельді жасаудың алғашқы кезеңдерінде басталады. Бұл параметрлерді факторлар деп атайды. Содан кейін сандық параметрлердің сандық мәндерінің мүмкін

мәндерін және сапалық (функционалдық) параметрлердің нұсқалары бөліп алынады. Оларды q деңгейі деп атайды. Бұл кезде алмасу саны

$N_s = q_1 \times q_2 \times ... \times q_k$,

мұнда к – фактор саны.

Егер факторлар саны көп болса, жүйенің зерттеудің толық емес факторлық анализ жоспарының әдістерінің бірі қолданылады. Мұндай әдістер тәжірибелерді жоспарлау теориясын егжей-тегжейлі жасалған.

Имтиациялық модельдеудің нәтижелерінің статистикалық сенімділігін қамтамасыз ете отырып, машиналық тәжірибенің ұзақтығын азайту әдістері тактикалық жоспарлау деп аталады.

Бір тәжірибенің ұзақтығына (модельдеу кезеңі T_m) жүйенің стационарлық дәрежесі, сипаттамалардың өзара тәуелділігі және тәжірибенің бастапқы шарттарының мәндері әсер етеді.

Тәжірибеде жиналған мәліметтерді белгілі бір сипаттамаларды өлшеуден тұратын уақытша факторлар деп қарастыруға болады, у сипаттамаларын өлшеу қатарын стохастикалық тізбектен алынған бөлік деп қарастыруға болады [50]. Егер бұл тізбек стационар болса, онда оның орташа мәні у уақытқа тәуелсіз болады. Уақытша y_1 ... y_n қатары $\ddot{y}_n$ шамасымен бағаланады. Эргодикалық тізбек үшін бұл бағалаудың дәлдігі N өскен сайын дәл болады.

Егер бағалаудың жіберілетін ең үлкен мәні (сенімділік интервалы) және у орташа шын мәні осы интервалда болу ықтималдылығы берілген болса, онда зерттелетін бөліктің ең кіші өлшемі болады. Ол өлшем тәжірибенің ең кіші ұзақтығына тең болады. Бірнеше сипаттаманы бағалау үшін модельдеу кезеңі ең үлкен мәнмен анықталады.

Имитациялық тәжірибеде статистикалық модельдеу кезінде шамалар жиыны әрбір шығыс сипаттамасымен өлшенеді. Ол бөліктерді келесі талдау мен пайдалануға ыңғайлы болу үшін өңдеу керек. Шығыс сипаттамалары көбінесе кездейсоқ шамалар және функциялар болғандықтан өңдеу математикалық күтуін, дисперсиясын және корреляциялық моменттерді есептеуден тұрады[50].

Машинада барлық өлшемдерді сақтау қажеттігінен құтылу үшін өңдеуді рекурентттік формуламен жасайды, ал бағалауды тәжірибе барысында жаңа өлшемдер пайда болған сайын өспелі қорытындылау әдісімен жүргізеді.

Стохастикалық сипаттамалар үшін салыстырмалы жиіліктер гистограммаларын -үлестірудің эмпирикалық тығыздығын тұрғызуға болады. Бұл мақсатта у сипаттамасының болжанған мәндері аймағын интервалдарға бөледі. Тәжірибе барысында өлшеулер кезінде сипаттамалардың әрбір интервалды болуы санын және өлшеулердің жалпы санын анықтайды. Тәжірибені аяқтаған соң әрбір интервал үшін сипаттамалардың болу санының жалпы өлшем саны мен интервал ұзындығына қатынасын есептейді.

Кездейсоқ стационар емес сипаттамалар үшін $T_м$ модельдеу кезеңі Δt бойынша кездейсоқ шамаларды бағалау сияқты өткізіледі.
Жүйенің сипаттамаларының параметрлері мен сыртқы әсерге тәуелділігін талдау үшін коррециялау дисперсиялық немесе регрессиялық әдістерді пайдалануға болады.

Корреляциялық талдау көмегімен екі немесе одан да көп кездейсоқ шамалардың арасындағы байланысты тағайындауға болады. Байланыстарда шамалардың арасында сызықтық байлансыс және оларды бірге үлестіру болғанда, корреляция коэффициентімен бағаланады.

Дисперсиялық талдауды [18,23,32,39,44] шығыс сипаттамаларына әр түрлі факторлардың салыстырмалы әсерін тағайындау үшін пайдалануға болады. Жеке компоненттердің мәндері бойынша талданатын шамаға жеке факторлардың әсерінің дәрежесі туралы қорытынды жасайды.

Тәжірибеде барлық факторлар сандық болған жағдайда сипаттамалар мен факторлардың арасында аналитикалық тәуелділікті табуға болады. Ол үшін регрессиялық анализ әдістері қолданылады.

Модельдеудің нәтижелерін талдауға модельдің оны параметрлеріне сезімталдығын талдау мәселесін де қосуға болады. Сезімталдықты талдау деп жүйенің жұмыс істеу сипаттамаларының параметрлерді мүмкін өзгерістеріне тұрақтылығын тексеруді айтуға болады.
Модельдеудің нәтижесі жүйенің жұмыс істеу қабілеттілігі туралы ең жаңа жобалау нұсқасын таңдауға немесе жүйені оңтайландыруға шешім қабылдауға пайдаланады. Жұмыс істеу қабілеті жөнінде жүйенің сипаттамалары параметрлерді рұқсат етілген өзгерістері кезінде тағайындалған щығу не шықпауына байланысты шешім қабылданады. Барлық жұмыс істеуге қабілетті нұсқалардың ішінен тиімділік критерийі ең үлкен мәні бары таңдалады[51]. Жүйені оңтайландыру ең жалпылау және күрделі болып келеді: жүйенің параметрлері мәндері жұмыстық жүктеменің көп мәндерінен тиімділік критерийін ең үлкен мәнін сипаттамаларын барлық n мәндерінде шектеуді сақтай отырып табу керек:

$$E_{opt} = \max(\min)E(y)$$

Егер y шығыс белгілі бір $f(y_i)$ үлестіру тығыздығымен берілген кездейсоқ шама болса, оңтайландыру есебіне төмендегідей стохастикалық шектеулер қою керек

$$y_{i_{\min}} \le y_i \le y_{i_{\max}}, \qquad i = 1,2,...,n$$

Мұндағы β_i нақты y_i шектеу аралығынан шығып кетпеуінің алынған ең кіші ықтималдылығы.

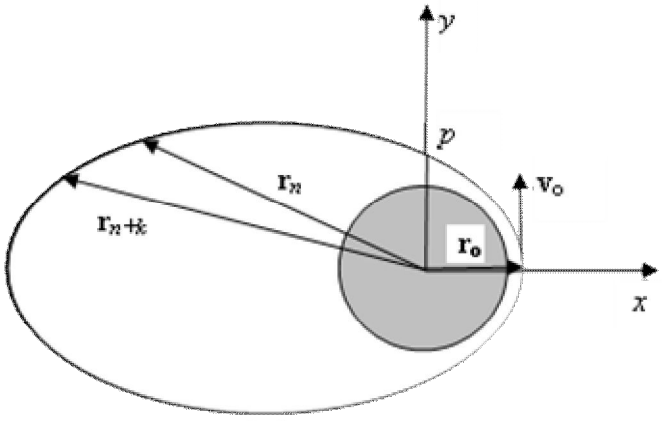
$$P\left(y_{i_{\min}} \le y_i \le y_{i_{\max}}\right) = \int_{y_{i_{\min}}}^{y_{i_{\max}}} f(y_i)dy_i \ge \beta_i .$$

2. Оқу есебінің қойылуы.

2.1. Тартудың центрлік өрісіндегі қозғалыс

Тартудың центрлік өрісінде қозғалыс туралы есеп белгілі бір жалпы физикалық заңдылықтарға бағынатын объектілердің қасиеттерін зерттеу үшін ДК пайдаланудың мүмкіндігін көрсетудің жақсы мысалы болып табылады. Оны тартудың центрлік өрісіндегі қозғалысын, серікті орбитаға шығару өрісі көбінесе екі этапқа бөлінеді. Бірінші этапта серік атмосферадан белгілі бір биіктікке жоғары көтеріледі. Содан кейін ракетасының соңғы сатысы қажетті горизанталь жылдамдық береді, әрі қарай ол инерциясымен қозғалады.

Кішкене дененің (серіктің) үлкен массалы центрдің (жерге) айналасында инерциялық ұшуын қарастырайық. Ол Жер бетіне жақын аралықта шеңбер траекторисымен қозғала отырып, оған құлап кетпеу үшін қандай ең кіші жылдамдықпен (бірінші космостық жылдамдық), қандай траекториямен қозғалысын, қандай жылдамдықпен ол тұйық емес траекториямен қозғалып Жерден кететіні (екінші космостық жылдамдық) қарастырайық. Сандық есепте, сол сияқты, Кеплер заңдарын: 1) тарту центрі орбитаның бір фокусында болатын 2) бағытталған радиус-вектор бірдей уақыт аралығында бірдей аудан сызатынын 3) тарту центрін серіктердің айналу уақытының квадраты олардың орбиталарының үлкен жарты осьтерінің қатынасындай болатынын тексеруге болады



2.1 Сурет. Координаталар жүйесі және серіктің мүмкінді траекториясы.

Модельдің негізіне бүкіләлемдік тартылыс заңын аламыз. Қарастырып отырған денеге тек тартылу күші әсер етеді деп есептейміз және Ньютон теңдеуін [54] жазамыз

$$m\mathbf{a} = -\frac{GMm}{r^3}\mathbf{r}\,. \qquad (2.1)$$

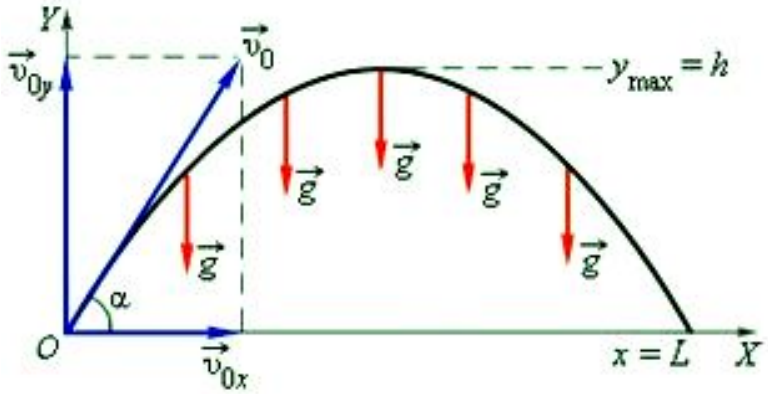
Мұнда m және M – серіктің массасы мен тарту центрінің массасы, G – гравитациялық тұрақты, r – серіктің тарту центріне қатысты орнын анықтайтын радиус-вектор, a – серіктің үдеуі. Бүкіләлемдік тартылыс заңы (2.1) түрінде материалдық нүктелер үшін жазылған болса да, ол сфералық-симметриялық денелер үшін де осындай формада болатынын ескертейік.
Центрлік күштің әсерінен болатын қозғалыс дененің бастапқы күйін және бастапқы жылдамдығы r_0 және υ_0 векторларына берілген бір жазықтықта болады. Тарту өрісінің центрінде Декарттық координаталар жүйесін және уақытты санау басын қозғалыс Х; У жазықтығында болатындай, қозғалыс басталарда жылдамдық Х осіне перпендикуляр болатындай етіп таңдай аламыз (2.1 сурет).
Бұл кезде бастапқы шарттарды төмендегідей етіп аламыз

$$t = 0: \quad x = x_0, \quad y = 0, \quad v_x = 0, \quad v_y = v_0 . \qquad (2.2)$$

(2.1) теңдеуі мен (2.2) шарттары серіктің траекториясын және оның барлық қасиеттерін толық анықтайды.
Компьютер модельдің негізінде:
- бастапқы шарттарға тәуелді дененің траекторияларының түрлерін зерттеуді жүргізу
- табылған шамалардың Кеплер заңдарына сәйкестігін тексеру қажет.

2.2.Нақты ортада көкжиекке бұрыш жасай лақтырылған дененің қозғалысы.

Нақты ортада көкжиекке бұрыш жасай лақтырылған материалдық дененің қозғалысын қарастырайық (2.2-сурет). Материалды дене m O нүктесінен көкжиекке ά бұрыш жасай υ_0 жылдамдықпен лақтырылсын. Дене оның жылдамдығына пропорционал Q кедергі күші әсер ететін нақты ортада қозғалсын. Лақтыру нүктесі және түсу нүктесі бірдей биіктікте орналасқан.



2.2-сурет. Көкжиекке бұрыш жасай лақтырылған дененің траекториясы.

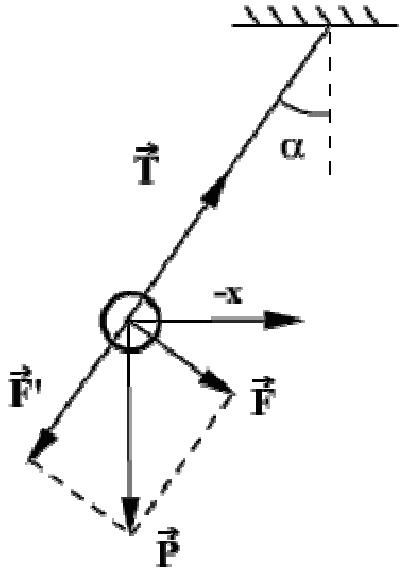
Қозғалыс уақытын, көтерілу уақытын, түсу уақытын, ең үлкен көтерілу биіктігін, ұшу аралығын, жоғары нүктедегі жылдамдығын, соңғы

жылдамдығын, соңғы жылдамдық пен OX осінің арасындағы бұрышты анықтау керек.

2.3.Физикалық маятник.

Физикалық маятниктің тербелістерін қарастырайық (2.3 сурет). Қозғалмайтын шарнирге ұзындығы lстерженнің шетінде массасы m жүк орналасқан маятник ілінген болсын.

Шарнирге үйкеліскен энергия жоғалмайтындай деп есептеуге болатындай тегіс деп есептелсін. Шарнирді қозғалмайды деп есептеу "стержень-шарнир" жүйесіне энергия келмейді, шарнир жүйеге жұмыс істей алмйды деген сөз.



2.3 сурет. Физикалық маятник "стержень-жүк" жүйесі.

Стержень салмақсыз және абсолют қатаң, яғни оның кинематикалық және потенциялық энергиясы нольге тең, ол жүк стерженнің осінің бойымен жылжи алмайды. Төмендегі шарттарды қоямыз: жүктің өлшемі стерженнің ұзындығымен араластырғанда аз (материалдық нүкте), еркін түсу үдеу g, ауаның кедергісін ескермейміз, тербеліс бекітілген вертикаль жазықтықта болады (бастапқы жылдамдық векторы осы жазықтықта орналасқан). Осы қарапайымдандыру алғы шарттарынан кейін маятниктің орны тек бір жалпыланған координата анықталатын түсінікті, оған стерженнің вертикальдан ауытқу ά (t) бұрышын аламыз.

Модельдеу барысында:

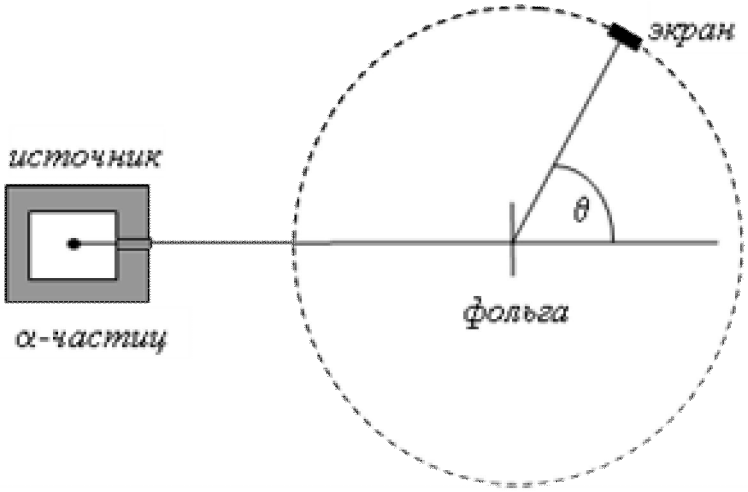
- қозғалыстың бұрыштық шамаларының уақытқа тәуелділік графиктерін тұрғызу;
- стерженнің вертикальдан ауытқу бұрышының сызықтық және сызықтық емес тәуелділіктерін есептеу;

- шарнирде үйкеліс болмаған және үйкеліс болған жағдайда жүйенің моделін есептеу және демонстрациялау;

-мәжбүрлеуші күшті ескере отырып физикалық маятниктің тербелістерін зерттеу қажет.

2.4. Атом моделідері .

Атом моделін 1903 жылы Дж. Томсон электронды ашқаннан кейін нейтрал аталды, оң зарядтардан басқа электрондар болатындығы анық болды. [54]. Дж, Томсон ұсынған атом моделі ішінде бөлменің ішінде жүзімдер сияқты электрондар орналасқан оң зарядталған сфера түрінде болды.



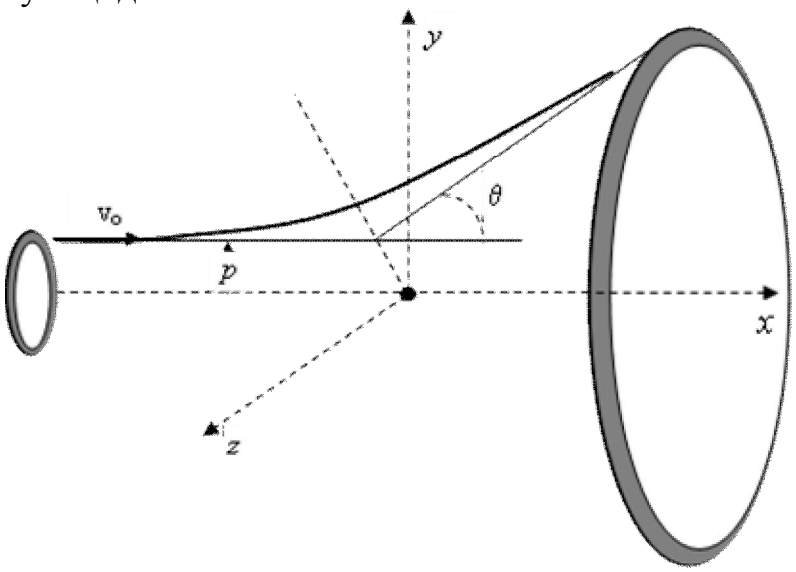
2.4-сурет. Бөлшектердің шашырауын бақылайтын қондырғы.

Атом шын мәнінде, құрылысын атомнан зарядталған бөлшектредің шашырауын бақылайтын тәжірибе арқылы анықтауға болады. Мұндай тәжірибелер Э. Резерфордтың басшылығымен жүргізілді. Зарядталған бөлшектер көзі ретінде ά – ыдырайтын (радий) радиоактивті препараттың кесегі пайдаланылады. Ά – ыдыраудың ерекшелігі ұшып шығатын ά бөлшектерді жылдамдықтары, іс жүзінде тең болады. Қорғасын диафрагмалардың көмегімен бөлініп алынған ά – бөлшектердің шоғы өте жұқа алтын фольгаға бағытталады. Фольгадан өткеннен кейін ά – бөлшектер алғашқы бағытына әр түрлі бұрыштармен шашырайды.

Шашырау бұрыштары флюоресценциялық экрандағы жарқылдарды микроскоппен бақылау арқылы тіркеледі. Тәжірибелік қондырғының схемасы 2.4 суретте бейнеленген. Шашыраған бөлшектердің үлестіру заңы шашырайтын бөлшектер мен шашырайтын атомдардың атомының өзара әсерлесу сипатымен анықталады, сөйтіп Резерфорд атомның құрылысын

түсінуге тырысты. Егер нейтрал атом тұтас шар (шартты түрде Томсон моделі дейік) болса, онда ά – бөлшек атомнан серпімді соқтығысу заңдары бойына серпіліп кетер еді. Егер нейтрал атомдар зарядтардың орналасуы біркелкі болмаса. ά – бөлшек атомға кіре алатын болса, атомның құрылысы бөлшектің шашырау заңында көрінеді.

Бөлшектердің шашырауы, шашырату центрінің маңайында олардың траекторияларының өзгерісі 2.5 суретте берілген. Шашырату центрі (алтын атомы) координаталар басында орналасқан. Сол жақты шексіз қашықтықта жылдамдықтары υ_0 бірдей, x осімен бағытталған біртекті бөлшектер көзі орналасқан. X осінен P арақашықтықта орналасқан бөлшек басында X осіне параллель түзу бойымен қозғалады, сосын атоммен әсерлесу нәтижесінде осы түзуден ауытқиды. Атомнан үлкен қашықтыққа кеткеннен кейін, бөлшек X осіне θ бұрыш жасай қайтадан бірқалыпты түзусызықты қозғалады. Осьтік симметрия нәтижесінде симметрия радиусы P, ені dp сақинадан шашыраған бөлшектер координата басында орналасқан қалыңдығы dθ болатын сақина ішінде болатын θ бұрышқа ауытқиды.



2.5 сурет. Координаталар жүйесі мен ауытқуы сұлбасы.

Егер бөлшектер көзі тығыздығы n , тұрақты жылдамдықпен қозғалатын бөлшектер ағынын шығаратын болса, экранға түсетін бөлшектердің шашырату центрлік dΩ денелік бұрышпен көрінетін бөлшек санын төмендегі формуламен есептеуге болады

$$dN = [n \cdot 2\pi p \cdot dp / (2\pi \sin\theta \cdot d\theta)] \cdot d\Omega . \qquad (2.3)$$

Шашыраудың дифференциалдық қимасы деп аталатын $s \equiv dN/(n \cdot d\Omega) = \left| p \cdot dp/(\sin\theta \cdot d\theta) \right|$ шамасын тәжірибее функциясы ретінде өлшеуге болады, ал сандық тәжірибеде α-бөлшек пен атомның әсерлесуінің берілген заңымен есептеуге болады. Осы нәтижелерді салыстыру өзара әсерлесудің дұрыс заңын таңдай алуға яғни атомның құрылысын анықтауға мүмкіндік береді.

Жылдам ұшатын бөлшектердің ауыр атомының ядросының кулондық электр өрісінде шашырауын қарастырайық. Біздің сандық тәжірибемізде өте жұқа алтын фольганың α-бөлшектерді шашырауын өлшеуге арналған Резерфордтың белгілі тәжірибесін модельдеу керек. Модельде координаталар басында орналасқан қозғалмайтын заряды ez ядро бар деп есептейміз. Ядродан қашықта x осі бағыттың p нысаналық параметр арақашықтығында U_0 жылдамдықпен α-бөлшек ұшып келеді. Бөлшекке әсер ететін күш центрлік болғандықтан, ол берілген ХУ жазықтығында векторларымен қозғалады. Шашырау θ бұрыш нысаналық параметрге тәуелді болады (2.5. суретті қара).

Бөлшектің қозғалыс теңдеулерін таңдай алған координаталар жүйесінің остеріне проекцияларымен жазамыз $t = 0: \quad x = -\infty, y = p, \mathrm{v}_x = \mathrm{v}_0, \mathrm{v}_y = 0.$

Есепті өлшемсіз шамалармен тұжырымдау үшін ұзындықтың, уақыттың және жылодамдықтың өлшем бірліктерін таңдап аламыз. Бір қарағанда ұзындыққа тән масштаб көрінбегендіктен оны *l* деп белгілейміз және қарапайымдылық шарттарына сай теңдеудің коэффициенттерін түпкілікті анықтаймыз. $dN = [n \cdot 2\pi p \cdot dp/(2\pi \sin\theta \cdot d\theta)] \cdot d\Omega.$ (2.3)

Есепке α-бөлшек көзінен шығатын оған тән U_0 жылдамдық бар. Бұл жылдамдық барлық тәжірибелерде бірдей, сондықтан оны жылдамдық масштабы ретінде алу ыңғайлы. Уақыт бірлігін ұзындық бірлігінің жылдамдық бірлігіне қатынасы түрінде анықтаймыз. Сонымен төмендегі қатынастарды аламыз:

$$m\frac{d\mathrm{v}_x}{dt} = \frac{2e \cdot eZx}{4\pi\varepsilon_0 r^3}, \quad \frac{dx}{dt} = \mathrm{v}_x,$$

$$m\frac{d\mathrm{v}_y}{dt} = \frac{2e \cdot eZy}{4\pi\varepsilon_0 r^3}, \quad \frac{dy}{dt} = \mathrm{v}_y,$$

Мұнда ~ белгісімен өлшемсіз шамалар белгіленген. Қозғалыс теңдеулері өлшемсіз шамалармен жазғанда төмендігедей түрге енеді:

$$\frac{d\tilde{\mathrm{v}}_x}{d\tilde{t}} = \frac{2e^2 Z}{4\pi\varepsilon_0 m\mathrm{v}_0^2 l}\frac{\tilde{x}}{\tilde{r}^3}, \quad \frac{d\tilde{x}}{d\tilde{t}} = \tilde{\mathrm{v}}_x, \quad \frac{d\tilde{\mathrm{v}}_y}{d\tilde{t}} = \frac{2e^2 Z}{4\pi\varepsilon_0 m\mathrm{v}_0^2 l}\frac{\tilde{y}}{\tilde{r}^3}, \quad \frac{d\tilde{y}}{d\tilde{t}} = \tilde{\mathrm{v}}_y, \qquad (2.4)$$

.

Енді l шамасын Ньютон теңдеуіндегі коэффициент 1-не тең болатын етіп алуға болады:

$$l = \frac{2e^2 Z}{4\pi\varepsilon_0 m \mathrm{v}_0^2}. \qquad (2.5)$$

Мұндай өлшемсіз шамалармен алғанда α-бөлшектің үдеуі және бастапқы шарттарының түрі (теңдеудегі барлық шамалар өлшемсіз және өлшемсіз шамалардың белгісі алынып қалған):

$$a_x = \frac{x}{(x^2+y^2)^{3/2}}, \quad a_y = \frac{y}{(x^2+y^2)^{3/2}}. \qquad (2.6)$$

$$t = 0: \quad x = -\infty,\ y = p,\ \mathrm{v_x} = 1,\ \mathrm{v_y} = 0. \qquad (2.7)$$

Зерттеулердің нәтижесі болып атап берілген екі моделдің дұрысыны таңдау болып табылады.

2.5. Қойылған есепті шешу және математикалық модельді құру

Есепті сандық талдау бірліктер ретінде есептің тән масштабын пайдаланып өткізу ыңғайлы [20;21]. Ұзындықтың бірлігі ретінде x_0 алған ыңғайлы. Егер Жер серігін қарастырсақ, онда бұл бірлік Жер радиусы R шамалас болады. R+h тең болады. Мұнда h серіктің Жер бетінен биіктігі. Енді кез-келген арақашықтық онда x_0 қанша рет санымен беріледі. Өлшемсіз x өлшенген x_0 ге бөлген метрмен өлшенген x-ке тең болады. Уақытты гравитациялық тұрақты және тарту сипаттамаларын пайдаланып құру ыңғайлы болады.

2.1. теңдеуінен GM/r^2 көбейтіндісі үдеудің өлшемімен (м/с^2) өлшенетінін көруге болады. Арақашықтық r-дің орнына x_0 алып, уақыт бірлігінен өлшенетін шаманы бейнелейік (с): $\left(GM/x_0^3\right)^{-1/2}$

Онда жылдамдық бірлігіне $x_0/\left(GM/x_0^3\right)^{-1/2}$, яғни $\left(GM/x_0\right)^{1/2}$ өрнегін алған жөн.

Осы бірліктермен өлшенген үдеудің проекциялары келесі теңдеулермен анықталады (осында және әрі қарай өлшемсіз физикалық шамалар үшін сәйкесті өлшемді шамалардың белгілеулері берілген):

$$a_x = -\frac{x}{(x^2+y^2)^{3/2}}, \qquad a_y = -\frac{y}{(x^2+y^2)^{3/2}}, \qquad (2.8)$$

Осыны уақыт бірлігі ретінде аламыз. Онда бастапқы шарттардың түрі төмендегідей болады: $t = 0: \quad x = -\infty,\ y = p,\ \mathrm{v}_x = \mathrm{v}_0,\ \mathrm{v}_y = 0.$

тың бірлігіне

$$t = 0: \quad x = 1,\ y = 0,\ \mathrm{v_x} = 0,\ \mathrm{v_y} = \mathrm{V0}. \qquad (2.9)$$

яғни $\mathrm{V0} = \mathrm{v}_0 \cdot \left(x_0/GM\right)^{1/2}$ өрнегін алған жөн.

Барлық физикалық шамалар енді салыстырмалы бірліктермен өлшенеді және барлық «серік-тарту центрі» жүйелерінің барлығы үшін бірдей

болады. Есептегі қалған тек V параметрі серіктің бастапқы кинетикалық және потенциялық энергия қандай қатынаста болатындығын көрсетеді. Шынында да серіктің бастапқы моменттегі кинетикалық энергиясы $K = m\mathrm{v}_0^2/2$ тең, ал Ньютон тартылу заңы үшін потенциялық энергия $\Pi = GMm/x_0$, и $\mathrm{V0}^2 = 2K/\Pi$ тең.

Кез келген уақыттағы серіктің жылдамдығының проекциясын және координаталық табу үшін уақыт осінде бір-бірінің кішкентай Δt интервалдық қашық орналасқан $t_и$ дискретті нүктелерді таңдап аламыз. Онда уақыт моментіне жылдамдықтың проекциялары уақыт моментінде келесі өрнектермен беріледі:

$$v_x^{(n+1)} = v_x^{(n)} + \Delta t \cdot a_x^{(n)}, \tag{2.10}$$

$$v_y^{(n+1)} = v_y^{(n)} + \Delta t \cdot a_y^{(n)}, \tag{2.11}$$

Осы уақыт моменттеріндегі координаталарды (Δt уақыт интервалы аз, жылдамдық интервалының соңында өзгермейді деп есептеп) бірқалыпты қозғалатындай есептейміз:

$$x^{(n+1)} = x^{(n)} + \Delta t \cdot v_x^{(n+1)}, \tag{2.12}$$

$$y^{(n+1)} = y^{(n)} + \Delta t \cdot v_y^{(n+1)}. \tag{2.13}$$

Бастапқы уақыт моменттерінде серіктің жылдамдық және координаталары белгілі:

$t_0 = 0: \quad x^{(0)} = 1, \quad y^{(0)} = 0, \quad v_x^{(0)} = 0, \quad v_y^{(0)} = \mathrm{V0}.$

(2.10)-(2.13) жүйесі Δt аз болғанда, біртіндеп серіктің траекториясын және басқа да сипаттамаларын анықтауға мүмкіндік береді.

Траекторияны классификациялау үшін оның е эксцентритетін есептеу ыңғайлы. Таңдап алынған координаталар жүйесінде эксцентритетті есептеу үшін x=0 болғанда y=p координатын анықтау керек болады. Сонда ℓ=[P-1]. Егер ℓ<1 болса, траектория тұйық сызық шеңбер, ℓ≠0 болғанда эллипс болады, ℓ=1 болғанда (сандық тәжірибеде бұл мәнді алу мүмкін емес), траектория парабола, ал ℓ>1 траектория гипербола.

Белгілі бір $T_k = t^{(n+k)} - t^{(n)} = k \cdot \Delta t$ уақыт ішінде радиус вектор сызатын ауданды төмендегідей жолмен есептейміз:

$$S_k = \tfrac{1}{2} \sum_{i=n}^{n+k} (x^{(i)} \cdot \mathrm{v}_y^{(i)} - y^{(i)} \cdot \mathrm{v}_x^{(i)}) \Delta t. \tag{2.14}$$

Траекторияның әр түрлі бөліктері үшін есептелген аудандарды салыстырған жөн. Кеплердің екінші заңы бойынша ол тең болуы керек.

Кеплердің үшінші заңы бойынша, айналу периодтығының квадраттарының қатынасы орбитаның үлкен жарты остерінің қатынасындай болады. Яғни

әрбір орбита үшін КЗ = T^2/a^3 тұрақты болып қала беруі керек. Эллипстің үлкен жарты осін тау үшін y=0 кезінде x-тің теріс координатасын табу керек. Бұл жағдай бірінші рет болған $t_м$ уақыт моменті серіктің айналу периодының жартысына тең болады. Осыдан эллипстің үлкен жарты осі $x_{мин}$ при $y = 0$, период $t_м$ екенін аламыз.

Нақты ортада көкжиекке бұрыш жасай лақтырылған дененің қозғалысы.

Нақты ортада көкжиекке бұрыш жасай лақтырылған дененің қозғалысын 6-есеп бойынша көрсеміз**.**

Декарттың ХОУ координаталар жүйесін x осі О және А нүктелері арқылы өтеді. Ньютон заңын координаталар осіне проекцияларын мына түрде жазуға болады:

$$m\frac{d^2x}{dt^2} = F_x, \quad m\frac{d^2y}{dt^2} = F_y, \tag{2.15}$$

Мұндағы F_x, F_y – материалдың нүктеге әсер ететін барлық күштердің координаталар осьтеріне проекциялары. Көкжиекке бұрыш жасай қозғалатын денеге екі күш әсер етеді. Ауырлық күші Ғ=mg дененің салмағына $F = P$ тең деп есептейміз, ал ортаның кедергі күші $Q = km\vartheta$, мұнда к – пропорционалдық коэффициенті. Қозғалыстың дифференциалдық теңдеулері (2.15)-тен алынады, ал өрнекке Р және Q күштерінің проекцияларының өрнектерін қойғаннан шығады:

$$F_x = P_x + Q_x, \; F_y = P_y + Q_y$$

Күштердің проекциялары:

$$P_x = 0, \; Q_x = -km\dot{x}, \; P_y = -mg, \; Q_y = -km\dot{y}$$

(2.15)-ке қойып көкжиекке бұрыш жасай лақтырылған дененің қозғалысының моделі болатын дифференциалдық теңдеулер жүйесін аламыз.

$$\begin{aligned} &\frac{d^2x}{dt^2} = -k\frac{dx}{dt}, \; \left.\frac{dx}{dt}\right|_{t=0} = v_0\cos\alpha, \; x(0) = 0, \\ &\frac{d^2y}{dt^2} = -k\frac{dy}{dt} - g, \; \left.\frac{dy}{dt}\right|_{t=0} = v_0\sin\alpha, \; y(0) = 0. \end{aligned} \tag{2.16}$$

Көбіне сандық шешу екінші реттік теңдеулер жүйесін (2.19) бірінші ретті дифферециалдық теңдеулер жүйесіне (теңдеуерді ретін төмендету) әкелген ыңғайлы:

$$\frac{dv_x}{dt} = -kv_x,\ \frac{dx}{dt} = v_x,$$
$$\frac{dv_y}{dt} = -kv_y - g,\ \frac{dy}{dt} = v_y, \qquad (2.17)$$
$$\frac{dz}{dt} = -g.$$

Бұл жағдайда бірінші тарауда келтірілген дискреттік теңдеулер, айырмалар тұрғызудың әр түрлі әдістерін пайдалануға болады.
Сақталу заңдары (қозғалыс интегралдары) дифференциалдық теңдеулер шешу кезінде алынған сандық шешуді тестілеуге көмектеседі, яғни ЭВМ-де шешудің дұрыстығын тексеру теңдеулері болып табылады. Сақталу заңдары дифференциалдық теңдеулерге қосымша болып келеді. Тәжірибеде нақты есепте бірнеше қозғалыс интегралдарды пайдалану керек, себебі пайдаланылған алгоритм дұрыс болмауы мүмкін. Берілген есепте ЭВМ жұмысын тексеру үшін артық айнымалы әдісін пайдаланамыз. Көмекші артық айнымалы Z(t) енгізіп, ол x(t), y(t) айнымалыларымен қатар бақылау теңдеуі ретінде алынған белгілі бір қатынасты қанағаттандыру керек. Бұл есепте бақылау қатынасының түрі

$$kx + \frac{dx}{dt} + ky + \frac{dy}{dt} - z = 0. \qquad (2.18)$$

Қосымша Z(t) артық айнымалысын алу әдісін (2.18)-ді уақыт бойынша дифференциалдап табуға болады. [24]. Сонда Z(t) үшін анықтаушы дифференциалдық теңдеу түрі:

$$\dot{z} = k\dot{x} + \ddot{x} + k\dot{y} + \ddot{y}. \qquad (2.19)$$

Шынында, (2.16)-ны (2.18)-ге қойсақ, онда Z бойынша туынды үшін өрнек аламыз. Көкжиекке бұрыш жасай лақтырылған дененің қозғалысын сипаттайтын дифференциалдық теңдеулер жүйесін (математикалық модельді) мына түрде жазуға болады:

$$\begin{aligned} &\dot{z} = -g, \\ &\ddot{x} = -k\dot{x}, \\ &\ddot{y} = -k\dot{y} - g, \\ &kx + \dot{x} + ky + \dot{y} - z = 0. \end{aligned} \qquad (2.20)$$

Есеп үшін бастапқы шарттардың түрі:

$$\begin{aligned} &z(0) = v_0(\cos\alpha + \sin\alpha); \\ &\dot{x}(0) = v_0\cos\alpha,\ x(0) = 0, \end{aligned} \qquad (2.21)$$

$$\dot{y}(0) = v_0 \sin\alpha\ ,\ \ y(0) = 0\ .$$

(2.20) теңдеулерін сандық шешу әдісі оларды айырмалық аналогтармен алмастыруда болады. Айырмалық теңдеулер жүйесі

$$\begin{aligned}
&\frac{v_{x,y+1} - v_{x,i}}{h} = -kv_{x,i}\,;\\
&\frac{x_{i+1} - x_i}{h} = v_{x,i}\ ,\ \ v_{x,0} = v_0\cos\alpha_0\ ,\ \ x_0 = 0\,;\\
&\frac{v_{y,i+1} - v_{y,i}}{h^2} = -kv_{y,i} - g\,;\\
&\frac{y_{i+1} - y_i}{h} = v_{y,i}\ ,\ \ v_{y,0} = v_0\sin\alpha_0\ ,\ \ y_0 = 0\ .
\end{aligned} \qquad (2.22)$$

көкжиекке бұрыш жасай лақтырылған дененің қозғалысының (2.20) математикалық моделінің дискретті аналогы болып табылады. Теңдеулердің айырмалық жүйесі

$$\begin{aligned}
&x_{i+1} = 2x_i - x_{i-1} - h^2 kx_{i-1}\ ,\\
&y_{i+1} = 2y_i - y_{i-1} - kh(y_i - y_{i-1}) - h^2 g\ .
\end{aligned} \qquad (2.23)$$

(2.16) теңдеулерінің дискретті аналогы болады. (2.22.-2.23) шеше отырып, біз дифференциалдық теңдеулердің дискретті модельдері болатын теңдеулерді шешеміз. Осы теңдеулердің шешулері бастапқы теңдеулердің шешулері деп аталады. Олар берілген теңдеулер жүйесінің шешуін қаншама қанағаттандырады, анығын айтқанда шешуіне жақындайды (апромаксицияланады) таңдап алынған сандық шешу әдісімен анықталады. Бұл есеп айырмалық сандық әдістер теориясында қарастырылады. [2.27]

Физикалық маятник.

Физикалық маятник 7-есеп бойынша көрсетеміз.Физикалық маятниктің ауырлық күшінің әсерінен тербелудің математикалық моделін Гамильтон принципінің негізінде алуға болады. [54]

Оның түрі

$$\frac{d^2\alpha}{dt^2} = -\frac{g}{l}\sin\alpha\ . \qquad (2.24)$$

$$\omega_0^2 = \frac{g}{l}\ .$$

белгілеуін енгізіп меншікті тербеліс жиілігі ұғымын енгіземіз.

Маятниктің тербелу теңдеуі (2.24) сызықтық емес екенін атап өтейік. Бұл «стержень-жүк» жүйесінің геометриялық күрделілігіне байланысты атап айтқанда: жүктің үдеуі координатасына Гук заңындағыдай пропорционал

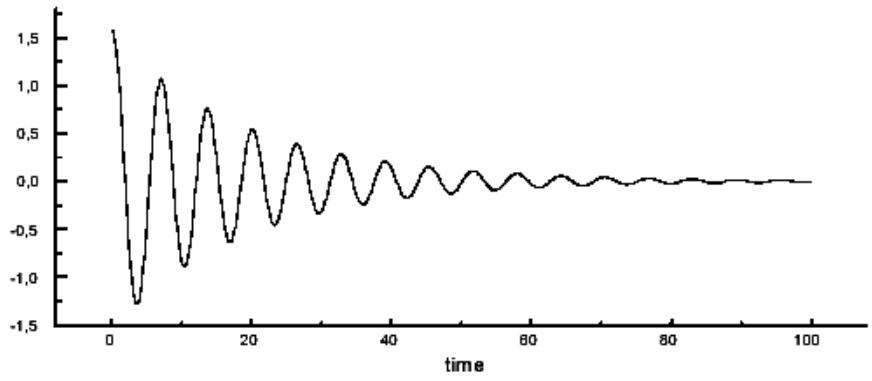
емес, тепе-теңдік күйінен ауытқуының (α-бұрышы) күрделі функциясы болады. Бұл ауытқулар аз болғанда, яғни sinα≈α, кішкентай тербелістер моделі сызықтық болады:

$$\frac{d^2\alpha}{dt^2} = -\omega_0^2\alpha\,, \tag{2.25}$$

Яғни математикалық маятниктің тербелісін сипаттайтын модель алдық. Егер шарнирдегі үйкелісті ескерсек, келесі теңдеуді аламыз:

$$\frac{d^2\alpha}{dt^2} = -\omega_0^2 \sin\alpha - \gamma\frac{d\alpha}{dt}\,, \tag{2.26}$$

Мұндағы γ-өшу коэффициенті тежеуші күштің өлшемін көрсетеді.



2.6. -сурет. Өшпелі тербелістердің амплитудасының уақытқа тәуелділігі

Физикалық маятниктің мәжбүрлеуші күштің әсерін ескергенде тербелісі төмендегідей болады:

$$\frac{d^2\alpha}{dt^2} = -\omega_0^2 \sin\alpha - \gamma\frac{d\alpha}{dt} + \frac{F(t)}{m}\,. \tag{2.27}$$

Теңдеудің ретін төмендете отырып сандық әдіспен шешетін теңдеулер жүйесін аламыз:

$$\frac{d\omega}{dt} = -\omega_0^2 \sin\alpha - \gamma\omega + \frac{F(t)}{m}\,,$$
$$\frac{d\alpha}{dt} = \omega$$

Сыртқы күшті гармониялық дпе қарастыруға болады.

$$\frac{F(t)}{m} = A_0 \cos(\omega_1 t),$$

Мұнда ω_1 – мәжбүрлеуші күштің бұрыштық жиілігі.
2.6. суретте (2.27) теңдеуін үйкелісті ескере отырып шешу мысалы келтірілген.

Атом моделі.

Дж.Томсон моделінде атом электрлі бейтарап ауыр бөлшек болып келеді. Егер ол α-бөлшек үшін мөлдір болмаса, тәжірибеде α-бөлшектің серпімді шашырауы бақылануы керек. Атомға дейін α-бөлшек түзудің бойымен тұрақты жылдамдықпен ұшады, атом бетінде серпімді шашырау болады; α-бөлшектің жанама құраушысы өзгермей қалады, ал нормаль құраушысы таңбасын өзгертеді. Атомның массасы бөлшектің массасынан көп үлкен деп есептейді. Одан әрі қарай бөлшектің қозғалысы түзу бойымен еркін жалғасады.

α-бөлшектің қозғалыс теңдеуін жазайық. Атомнан үлкен қашықтықта α-бөлшек ϑ_0 жылдамдықпен x осі бағытында p-нысаналық арақашықтықта қозғалсын. Бөлшек әр уақытта да 2.5.- суретте x, y, деп белгіленген бір жазықтықта қозғалады. Θ шашырау бұрышы p нысаналық параметрге тәуелді. Теңдеулерді өлшемсіз шамалар түрінде жазу үшін арақашықтық өлшемінің бірлігіне атомның радиусы a, жылдамдық бірлігіне α-бөлшектің бастапқы уақыт моментіндегі жылдамдығы ϑ_0 уақыт бірлігіне a/ϑ_0 алайық. Бөлшектің жылдамдығын, уақыттың t_n – тізбектелген моментінде орнын есебін координаталар осінде проекциясынды есептеуді төмендегідей тұжырымдауға болады:

$$t_0 = 0: \quad x^{(0)} = X0, \quad y^{(0)} = p, \quad v_x^{(0)} = 1, \quad v_y^{(0)} = 0;$$
$$t_{n+1} < T_{столк}: \quad x^{(n+1)} = x^{(n)} + dt, \quad y^{(n+1)} = p, \quad v_x^{(n+1)} = 1, \quad v_y^{(n+1)} = 0;$$
$$t_{n+1} = T_{столк}: \quad x^{(n+1)} = -\sqrt{1-p^2}, \quad y^{(n+1)} = p, \quad v_x^{(n+1)} = p^2 - x^{(n+1)} \cdot x^{(n+1)}, \quad v_y^{(n+1)} = 2px^{(n+1)};$$
$$t_{n+1} > T_{столк}: \quad x^{(n+1)} = x^{(n)} + v_x^{(n)} dt, \quad y^{(n+1)} = y^{(n)} + v_y^{(n)} dt, \quad v_x^{(n+1)} = v_x^{(n)}, \quad v_y^{(n+1)} = v_y^{(n)}.$$

Бөлшек пен атомның кішкене dt шамасында соқтығысу моментін $x^{(п)} \cdot x^{(п)} + y^{(п)} \cdot y^{(п)} \leq 1$ шартынан анықтауға болады.
Траекторияны толықтай құрап болғаннан кейін θ шашырау бұрышын, S параметрін есептеп шығаруға болады. Оны есептеу p-нысаналық параметрге жуық нысаналық параметрмен траектория тұрғызылады (төменде келген алгоритмде Δp=0,0001 деп алынды).Әртүрлі уақыт моментінде α-бөлшектің жылдамдығының проекциясын және координатасын есептеу үшін уақыт осінде бір-бірінен қашықтығы Δt кішкентай интервалы болатын $t_п$ дискретті нүктелерді таңдап аламыз. Онда жылдамдық траекториялары $\upsilon_x^{п+1}$ және υ_y^{y+1} $t_{п+1}$ уақыт моментінде (бұл

интервалда үдеу өзгермейді деп есептейміз) төмендегі өрнектермен келтіріледі:

$$v_x^{(n+1)} = v_x^{(n)} + \Delta t \cdot a_x^{(n)}, \qquad (2.28)$$

$$v_y^{(n+1)} = v_y^{(n)} + \Delta t \cdot a_y^{(n)}, \qquad (2.29)$$

Ал осы моменттегі координаталар да бірқалыпты қозғалыс кезіндегідей (Δt интервалы кішкентай, осы интервалдағы жылдамдығы интервалдың соңындағыдай деп есептей отырып) есептеледі:

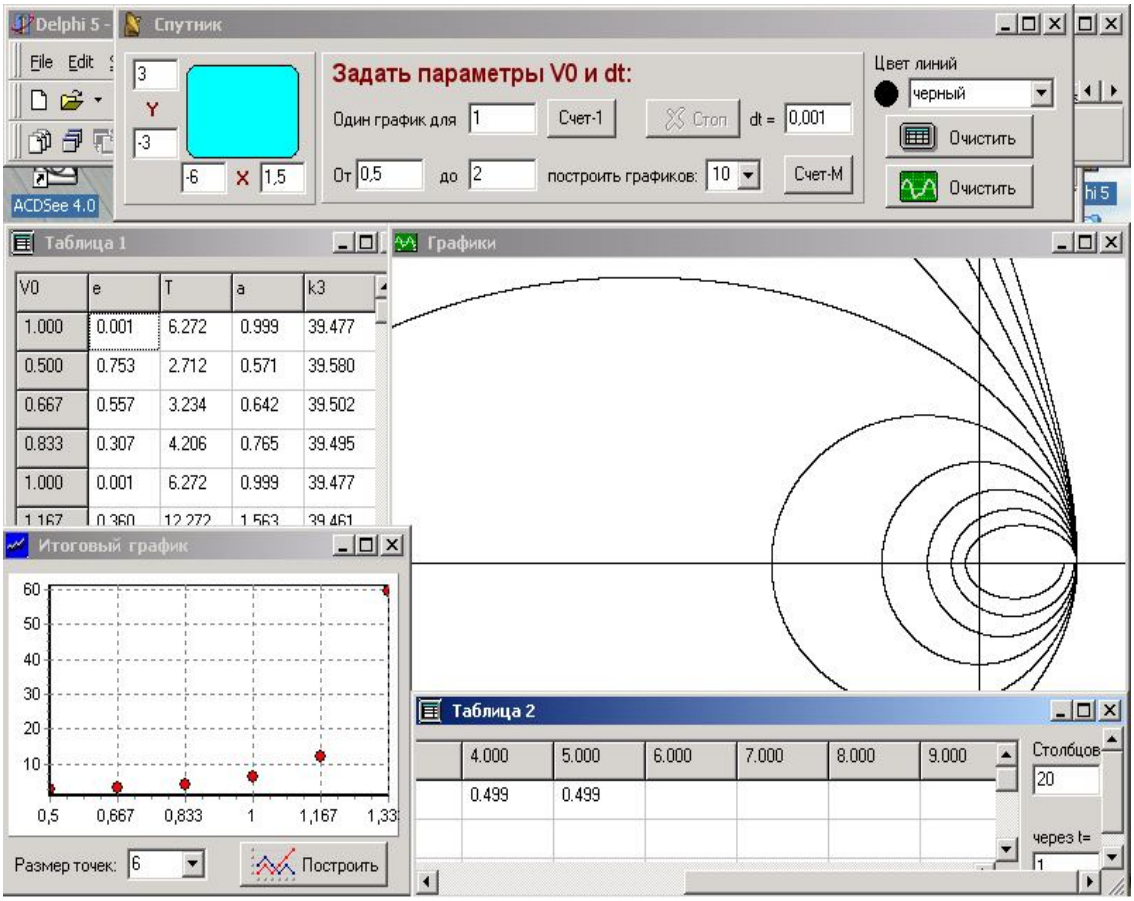
$$x^{(n+1)} = x^{(n)} + \Delta t \cdot v_x^{(n+1)}, \qquad (2.30)$$

$$y^{(n+1)} = y^{(n)} + \Delta t \cdot v_y^{(n+1)}. \qquad (2.31)$$

Бастапқы уақыт моментінде α бөлшектің жылдамдығының проекциялары мен координаталары белгілі:

$t_0 = 0: \quad x^{(0)} = -X_0, \quad y^{(0)} = p, \quad v_x^{(0)} = 1, \quad v_y^{(0)} = 0.$

(2.28)-(2.29) жүйесі Δt аз болғанда біртіндеп α бөлшектің траекториясын және қандай бұрышқа ауытқуын жеткілікті дәлдікпен есептеуге болады α-бөлшектің бастапқы жағдайын беретін X_0 тұрақтысы жеткілікті үлкен болуы керек. Сандық тәжірибеде оның мәні 20 тең деп алынады. Шашырау бұрышы есептелген арақашықтықта осы шамада алынады.



Виртуальды зертханалық көрініс

2.6. Delphi -7 визуальды программалау

Delphi ортасында жұмыс істеу технологиясы объектіге бағытталған программалау және визуальды программалау идеясына негізделген. Объектіге бағытталған программалау идеясы деректерді инкапсуляциялау (біріктіру) және оларды өңдеу құралдарын (әдістерін) типке классқа біріктіру болып табылады. Нақты класс айнымалысы объект болып табылады. Объект мысалдары ретінде терезені басқару элементтері: батырмалар, тізімдер, мәтіндік өрістер және т.б. Delphi визуальды программалау ортасы Pascal тілінің объектіге бағытталған нұсқасына орнатылған автоматты графикалық қоршам. Pascal тілінің құрылымдық бірлігі ретінде деректер мен бұйрықтар болып табылса, Delphi ортасында ондай құрылымдық бірлік компонент болып табылады.

Пішін (Форма) деп Windows терезесінің қасиеттері бар және басқа компоненттер орналастыру үшін қолданылатын компонент. Формадағы компоненттер көрінетін және көрінбейтін болуы мүмүкін. Оның біріншісі қолданушымен сұхбат ұйымдастыру үшін қолданылады. Оларға әр түрлі батырмалар, тізімдер, жолдық өрістер, суреттер жатады. Олар программаның орындалу барысында бейнеленеді. Көрінбейтін компоненнтер компьютердің жүйелік ресурстарына рұқсат алу үшін қолданылады.

Компоненттер палитрасы. Компоненттер палитрасы бас менюде орналасқан және ол көпбеттік блокнот түрінде болады. Әрбір беттің өз компоненттер жиыны болады. Компонентті терезе ортасына орналастыру үшін, оның пиктограммасына екі рет шерту керек. Егер компонентті пішіннің кез-келген жеріне орналастыру керек болса, онда оның пиктограммасына бір рет шертіп, одан кейін пішіннің керек жеріне бір шертсе жеткілікті. Егер де бір компонентті бірнеше рет қою керек болса, онда Shift пернесін баса отырып оның пиктограммасына шерту керек – содан кейін ғана пішін бетіне шерту керек. Бұл режимнен шығу үшін компоненттер палитрасындағы стрелка белгісіне шерту керек.. Таңдалған компонентті пішін бетінде жылжытуға, сонымен қатар оның өлшемін маркерді соза отырып өзгертуге болады.

Объектілер инспекторы. Объектілер инспекторы көмегімен объект қасиеттерінің алғашқы мәндерін және стандартты оқиғаларға сәйкес әрекеттерін беруге болады. Объектілер инспекторы ағымдағы форманың компоненттер тізімінен және екі бетбелгіден тұрады: қасиеттері (Properties) және оқиғалар (Events). Объектілер инспекторын активтеу үшін F11 пернесін басу керек. Осы терезені қарастырайық .

Қасиеттер бетбелгісі екі бағаннан тұрады: сол жағы компоненттер қасиеттерінің атауларынан, ал оң жағы – олардың мәндерінен тұрады. Қасиеттері қарапайым және құрама болуы мүмкін. Құрама қасиеттері

басқа да қасиеттерден тұрады. Ондай қасиеттер объектілер инспекторында “+” белгісімен белгіленген, мысалы, +Font.

Оқиғалар бетбелгісі де екі бағаннан тұрады. Сол жақ бағанда стандартты оқиғалар атауы, ал оң жақ бағанда - әдістер атауы (процедуралар) бейнеленеді. Әрбір стандартты оқиғаға әдіс атауы жауап береді, ол оң жақ бағанға тышқанды екі рет шерткенде пайда болады. Осы кезде программа мәтіні терезесіне әдіске сәйкес процедура шаблоны қосылады. Шаблонды сәйкес бұйрықтармен толтыру керек.

Сандық және мәтіндік типті (Width, Name) қасиеттерді енгізу үшін стандартты енгізу қрістерін қолданамыз. Саналатын типті қасиеттердің мәндері (Align, Cursor) аралас тізіммен сипатталады, олардың ішінен қажеттісі таңдалады. Кейбір құрама қасиеттер (Font, Picture, Glyph) сұхбат терезесін қолданады.

Форма беті. Форма ол құрылатын программамыздың негізгі объектісі. Ол Windows интерфейсінің құралы. Форманың ішкі кеңістігі жұмыс облысы деп аталады. Жұмыс облысына оған компоненттерді ыңғайлы орналастыру үшін туралау торы сызылған. Егер компонент жолынан бір объектіні шертсек, ол белгіленеді. Форма бетінде шертсек, сол объект формада пайда болады. Енді форманың басқа жерінде, формада орналасқан компонент фокусты формаға қайтарады да белгіленбей көрінеді. Пайда болған объектінің орнын көлемін жүгіртпе курсорымен өзгертуге болады. Ол үшін оны шертіп, қайта белгілеу керек. Енді объект жақтарынан ұстап тартып көлемін өзгертеміз. Ал объект бетінде сол батырмамен шертіп, жібермей, басқа жерге апарып жібере салсақ, компонент сол жерге орнын ауыстырады. Белгіленген объектіні жою үшін Delete пернесін, немесе Edit менюсындағы осы команданы шерту керек.

Проект құрылымы. Проект деп Delphi-дегі файлдар жиынын айтамыз. Бұл файлдардан Delphi орындауға дайын программа құрады. Проек құрамына міндетті түрде мына файлдар кіреді:

- Проек файлы *.dpr. Бұл көлемі жағынан аса үлкен емес файл, онда барлық проект файлдарына сілтемелер жазылған. Программа коды Object Pascal тілінде жазылған. Осы файл программаны инициализациялайды, яғни іске қосады.
- Проектіге кіретін барлық формаларды сипаттау файлы: модуль файлы *.pas және форма файлы *.dfm. Проектінің әрбір формасына өз модулі сәйкес келеді.
- Программа ресурстарыны файлы *.res. Онда формаға кірмейтін ресурстар сипатталған, мысалы программа пиктограммасы.
- Проект параметрінің файлы *.dof.
- Орта параметрлерінің файлы *.drf, *.dsk, *.dsm. Бұл файлдар тек компиляциядан кейін ғана пайда болады.

Delphi-проектіні сақтау үшін модуль атауларын және проект атауын беру керек. Delphi-проектіні көшіру үшін міндетті түрде *.dpr, *.dfm, *.pas, *.res типті файлдарды көшіру керек, қалғандары автоматты түрде құрылады .

Кодтар редакторы. Кодтар редакторы жеке терезеде орналасқан. Бұл терезе көпбеттік блокнот түрінде ұйымдастырылған. Жаңа проект ашылған кезде Form1 формасына сәйкес Unit1.pas модуліне осы форманың сипаттамасына сәйкес программа коодын жазады. Unit1.pas модулінің түрі мынадай:

```
unit Unit1; { Модуль атауы }
interface   { интерфейс бөлімі, яғни процедуралар мен функцияларды хабарлау }
uses    {қосымша қолданылған модульдер тізімі  }
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type { объектілер типі мен класс сипаттамасы }
  TForm1 = class(TForm)
  private   { класстың іш бөлігі }
    { Private declarations }
{ Мұнда осы форма класына кіретін тұрақты, айнымалыларды, функция, процедураларды жариялауға болады, олар басқа модульдерден көрінбейді }
  public  { класстың ашық бөлігі }
    { Public declarations }
{ Мұнда басқа модульдерден көрінетін осы форма класына кіретін тұрақты, айнымалыларды, функция, процедураларды жариялауға болады }
  end;
var  { глобальды айнымалыларды жариялап, сипаттау бөлігі}
  Form1: TForm1;
{ қолданушы процедуралары мен функцияларын жариялау  }
procedure Information;
procedure SetPicture;
{ Процедуралар мен функцияларды хабарлау бөлігі }
implementation
{$R *.DFM}    { Форманы сипаттау файлы қосылады }
procedure Tform1.Button1Click(Sender: Tobject);
   begin
   { Мұнда қолданушы процедура денесін жазады  }
   end;
end. { модульдің соңы }
```

Delphi-де дайындалатын программа проект деп аталады. Форма программаны дайындау алдында ашылатын, программаның сұхбаттық терезесі. Алғашқыда ол Form1 атауымен көрінеді. Проект құру үшін формаға компоненттер палитрасында орналасқан түрлі компоненттер

орнатылады. Форманың және формаға енгізілетін компоненттердің түрлі қасиеттері бар. Қасиет – айнымалылардың ерекше түрі. Олар объектінің түрлі мүмкіндіктерін сипаттап, ағымдық күйін анықтайды. Программа құру форманы не онда орнатылған кейбір қасиеттерін өзгертуден басталады. Форманың негізгі қасиеттері:

Name (Атау) – формаға берілген атау. Ол Delphi объектілерінің негізгі қасиеттерінің бірі. Программаның жұмыс істеуі барысында Delphi объектіні осы атау бойынша ажыратып таниды. Delphi-дің формаға автоматты түрде алғашқы рет меншіктеген атауын (Form1) өзгертіп, басқа атау беруге болады. Ол үшін қасиеттер терезесінен Name қасиеті таңдалып, жаңа атау пернетақтадан енгізіледі.

Font (Шрифт) – формаға шығарылатын мәтін шрифтінің қасиеті. Оны таңдап, оң жағында көрінген көп нүкте (…) түймесін шерткен кезде Windows-тың сұхбат **Шрифті таңдау** терезесі көрінеді. Терезеден қажетті шрифт типін, өлшемін таңдап **ОК** түймесін шерту керек.

Caption (Тақырып) – форма терезесінің тақырыбына енгізілетін мәтін.

Color (Түс) – форманың түсін орнату қасиеті. Ол таңдалған кезде оң жағында тілсызық түймесі көрінеді. Тілсызық түймесі - қасиет мәнінің бірнеше екенінің белгісі. Тілсызық белгісін шерткен кезде мәндер (түстер терезесі ашылады. Тізімде көрінген қалаған түс таңдалған соң форма сәйкес түске боялады.)

Width (Ен), **Height** (биіктік) – пиксель, өлшем бірлігімен берілген форманың ені мен биіктігін орнату қасиеттері (бұл мәндер форманы қолдан кеңейту не сығу кезінде де автоматты түрде орнатылып қойылады).

Компоненттер палитрасында орналасқан көптеген компоненттерге осы қасиеттер тән.

Delphi-де программалар түрлі оқиғалар арқылы басқарылады. Оқиға – программаның жұмыс істеуі барысында объект жағдайының белгілі бір әрекетке жауап ретінде өзгеруі. Мысалы, пайдаланушы программа құру үшін алдымен формаға компонент орнатуы, форманы не формада орналастырылған компонентті тышқан арқылы шертуі мүмкін. Оның әр іс-әрекеті оқиға шақырады.

Delphi-де әр оқиғаға атау беріліп қойылған. Мысалы, компоненттер палитрасының Button түймесі арқылы формада орнатылған Button1 компонентін шерту Click (Шерту) оқиғасын шақырады.

Әр объектіге байланысты оқиғалар көп. Олар қасиеттер терезесінің Events қосымша бетіне енгізілген. Терезеде оқиға атауларының алдына On префиксі (қосымшасы) енгізіліп жазылған. Ол атаудың оқиға екендігін білдіретін белгі.

Delphi ортасында жиі кездесетін оқиғалар:

OnClick – тышқан батырмасын бір рет басу;

OnDblClick – тышқан батырмасын екі рет басу;

OnKeyDown – пернені бас;

OnCreate – форманы екі рет шерту;
OnActivate – форманы активті қалыпқа келтіру.
Оқиғаға байланысты құрылатын процедураны оқиға өңдеуші не оқиғаны өңдеу процедурасы деп аталады. Процедура дайындамасының жазылу түрі:

```
Procedure <атау> (Sender:Tobject);
   <сипаттау бөлімі>
  begin
   <процедура денесі>
  end;
```

мұндағы Sender - құрылатын процедураның қай класқа тиістілігін анықтайтын параметр.

Әрбір практикалық жұмыстың басында сол жұмысқа қажетті компоненттерді және олардың негізгі қасиеттерін кесте түрінде келтіріп отырамыз. **Form** объектісі программа терезесін құру үшін қолданылады. Оның негізгі қасиеттерін қарастырамыз:

Кесте-1. **Form** объектісі программа терезесін құру үшін негізгі қасиеттер

Қасиеттері	Қасиеттердің сипаттамасы	қабылданатын мәндерінің мысалы
ActiveControl	Формадағы активті объектінің есебі үшін	Button1, Edit2
AutoScroll	Формадағы сызғыш белгілерінің бар болуы	True, False
BorderStyle	Терезе өлшемін өзгерту мүмкіншілігі	bsSizeable(өлшемі кез-келген терезе), bsDialog, bsNone(өлшемі бекітілген терезе)
Width, Height	Терезенің ұзындығы мен биіктігі пиксельмен беріледі	503, 224 (сандық мәндер)
Font	Шрифт	құрама қасиет, сұхбат терезесінде беріледі
HorzScrollBar VertScrollBar	Сызғыш белгілерінің параметрлері	құрама қасиет
Icon	Программаның орындалуы барысында форма тақырыбында орналасқан пиктограмманы беру	(None) – Delphi-дегі стандарт пиктограмма, немесе белгілі бір *.ico файлдан жүктелген
Name	Форма атауы	Form1 (идентификатор)
Caption	Форма тақырыбы	Кезкелген символдар жолы
Color	Форма фонының түсі	ClGreen, clInfoBk(саналатын тип) $004525B1(сандық мән-

		сұхбат терезесінде беріледі)
Cursor	Орындалу кезеңіндегі терезенің бос орнындағы курсор түрі	CrDrag, crCross, crHelp, crArow (саналатын тип)
Enabled	Орындалу барысында формадағы объектілерге амал орындаудың рұқсаты.	True, False
Left, Top	Терезенің сол жақ жоғарғы бұрышының координатасы пиксельмен беріледі	200, 108 (сандық мәндер)
Position	Программаны орындауға жіберген кездегі терезенің өлшемі мен орны	poScreenCenter, poDesigned
WindowsState	Программаны орындау кезіндегі терезенің қалпы	wsNormal, wsMaxomized, wsMinimized

Жолдық өріс **Label** программа терезесінде текст құру үшін қолданылады. Жоғарыда аталған қасиеттермен қатар оған мынадай қасиеттер де тән.

Кесте-2. **Label** программасының қасиеттері

Қасиеттері	Қасиеттердің сипаттамасы	қабылданатын мәндерінің мысалы
Align	Орналасқан объектіге - формаға- қатысты тураланады	alButtom, alClient, alLeft, alNone, alTop
Alignment	өріс шетіне қатысты текстті туралау	taCenter, taLeftJustify, taRightJustify
AutoSize	өріс шетін текст шетіне келтіру	True, False
Visible	Объектінің көрінуі	True, False
WordWrap	Текст сөздерін жаңа жолға көшіру	True, False

Берілгендерді енгізу шығаруды ұйымдастыру. InputBox функциясы. Show Message**.**

InputBox функциясы.

Енгізу терезесі Delphi-дің стандартты **InputBox** функциясының терезесі. Программада InputBox функциясын пайдалану командасының жазылу үлгісі:

<айнымалы>:=InputBox(‘<тақырып>’,’<түсініктеме>’,’<мән>’);

Мұндағы *айнымалы* – мәні функция терезесіне енгізілетін жолдық типті айнымалы атауы (InputBox) функциясының мәні әркезде жолдық (String)

типті. Мән меншіктелетін айнымалы (х) программада х: string түрінде сипатталуы тиіс;
тақырып – енгізу терезесінің тақырыбы ретінде жазылатын мәтін;
түсініктеме - енгізу терезесінің ішінде жазылатын түсініктеме мәтін;
мән – функция терезесі көрінген кезде оның енгізу өрісінде көрінетін мәтін. Әдетте оны бос етіп қалдырады. Мысалы, программада х:=4.7 меншіктеу командасын InputBox функциясын пайдаланып, мынадай түрде беруге болады:
х:=InputBox(‘Аргумент мәні‘,’х=’,’’);
Команданың орындалу барысында енгізу терезесі шығады, сонда х-тің мәнін енгізіп ОК кнопкасына шерту керек. Ол меншіктелетін айнымалы (х) жолдық типті етіп қабылданғандықтан, стандартты типті түрлендіру функцияларын пайдаланып, оны сандық типті етіп түрлендіруге болады.

Стандартты типті түрлендіру функциялары.

Стандартты типті түрлендіру функцияларын кесте түрінде келтірейік:

Кесте -3Стандартты типті түрлендіру функцияларын

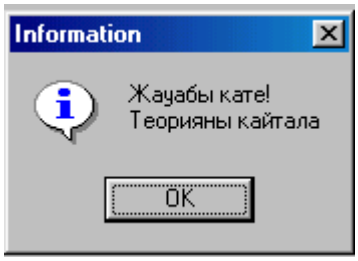
Функция	Орындайтын іс-әрекеті
StrToFloat(x)	Жолдық типті х айнымалысын нақты санға түрлендіру
FloatToStr(x)	х нақты санын жолдық типті етіп түрлендіру
FloatToStrF(x,f,s,o)	х нақты санын форматты жолдық типке түрлендіру мұндағы f-кескіндеу форматы. Ол көбінесе ffGeneral не ffFixed түрінде жазылады. s-барлық цифрлар саны – дәлдік о-ондық нүктеден соң жазылатын цифрлар саны – ондық дәлдік. Мысалы, FloatToStrF(x,ffFixed,7,3);
StrToInt(x)	Жолдық типті х айнымалысын бүтін санға түрлендіру
IntToStr(x)	х бүтін санын жолдық типті етіп түрлендіру
StrToDate(x)	Жолдық типті х айнымалысын датаға түрлендіреді
DateToStr(x)	х дата типті айнымалыны жолға түрлендіреді
StrToTime(x)	Жолдық типті х айнымалысын уақытқа түрлендіреді
TimeToStr(x)	х уақыт типті айнымалыны жолға түрлендіреді

Шығару функциялары.

Delphi-де нәтижені ShowMessage процедурасының терезесіне шығаруға болады. Процедураның жазылу үлгісі:
ShowMessage(s);
Мұндағы
s – жолдық типті өрнек. Егер ол сандық тпті болса, онда түрлендіру функцияларын қолдану керек.

Мысалы,ShowMessage(FloatToStr(s));
немесе ShowMessage(FloatToStrF(s,ffgeneral,7,3));

Ал MessageDlg функциясы хабарлама түріндегі терезеге стандартты ескерту белгілерін шығаруға, командалық кнопкалар саны мен типін көрсету мүмкіншілігін береді. MessageDlg функциясының мәні сандық.
Мысалы
s:=MessageDlg('Жауабы қате!'+#13
+'Теорияны қайтала',mtinformation,[mbOK],0);
инструкциясын орындағандағы терезе түрі:



Жалпы жағдайда MessageDlg функциясын шақыру үлгісі:

S:=MessageDlg(хабарлама, тип, кнопкалар, түсініктеме контексті)

Мұндағы
Хабарлама – шығарылатын текст;
Тип – хабарлама түрі. Ол ақпараттық, ескерту немесе ерекше қате туралы хабарлама түрінде болады. Әрбір хабарлама түріне белгілі бір белгі сәйкес келеді. Хабарлама типі атаулы тұрақты түрінде беріледі. (Кесте 3.1.)
Кнопкалар – хабарлама терезесінде бейнеленетін кнопкалар тізімі. Тізім бірнеше атаулы тұрақтыдан тұруы мүмкін, олардың арасына үтір белгісі қойылады. Ал жалпы тізім квадрат жақшаға алынады. (Кесте 3.2.) Мысалы, хабарлама терезесінде OK және Cancel кнопкалары пайда болуы үшін кнопкалар тізімі мынадай болуы керек:
[mbOK, mbCancel]
Түсініктеме контексті – түсініктеме жүйені анықтайтын параметр, ол көрсетілсе қолданушы <F1> - ді басқан кезде түсініктеме шығады, ал көрсетілмесе Түсініктеме контексті параметрінің мәні нолге тең.
MessageDlg функциясы қайтаратын мәндер қолданушының қай кнопканы басқандығын анықтауға мүмкіндік береді. (Кесте-4.)

Кесте-4. MessageDlg функциясы

Тұрақты	Хабарлама типі	Белгі
MtWarning	Ескерту	Ішінде леп белгісі (“!”) бар үшбұрыш белгі
MtError	Қате	Ортасында кресті (“x”) бар дөңгелек

		белгісі
MtInformation	**Ақпарат**	Ортасында “i” белгісі бар дөңгелек белгі
MtConformation	Растау	Ортасында “?” белгісі бар дөңгелек белгі
MtCustom	Әдеттегі	Ешқандай белгі жоқ

Тұрақты	Кнопка	Тұрақты	Кнопка
MbYes	Yes	mbAbort	Abort
MbNO	No	mbRetry	Retry
MbOK	OK	mbIgnore	Ignore
MbCancel	Cansel	mbAll	All
MbHelp	Help		

Функцияның мәні	Диалог сәйкес кнопкаларды басқанда аяқталды
MrAbort	Abort
MrYes	Yes
MrOk	Ok
MrRetry	Retry
MrNo	No
MrCancel	Cancel
MrIgnore	Ignore
MrAll	All

Берілгендерді форма бетінен енгізу.

Берілгендерді енгізу үшін компоненттер палитрасының *Standard* бетінде орналасқан **Edit** компонентін қарастырамыз. **Edit** компоненті пернеден символдарды енгізу үшін қолданылады. Егер енгізілетін мән сандық мән болса, онда түрлендіру функцтияларын пайдаланып түрлендіру керек. **Edit** компонентінің алдында айтылған қасиеттерден басқа мынадай қасиеттері болады:

Кесте-5. **Edit** компонентінің қасиеттері

Қасиеттері	Қасиеттердің сипаттамасы	қабылданатын мәндерінің мысалы
CharCase	Енгізу өрісіне егізілетін символ түрі	EcNormal(әдеттегі), ecUpperCase(кіші әріптер), ecLowerCase(бас әріптер)
CtI3D	Объектінің көлемді бейнеленуі	True, False
PasswordChar	Пароль енгізуге арналған символ	#0(мәтіннің әдеттегі бейнеленуі), *(мәтін жұлдызша түрінде

		бейнеленеді), 0(мәтін нольдермен бейнеленеді)
ReadOnly	Мәтінді өзгерту мүмкіншілігі	True (мәтінді өзгертуге болмайды), False (мәтінді өзгертуге болады)
Hint	Мышқан жүгіртпесін апарғанда пайда болатын көмекші мәтін	“ Қосындыны енгіз“(кез-келген жолдан тұратын символ)
ShowHint	Көмекшіні көрсету/көрсетпеу	True, False
Text	Түзету өрісіндегі мәтін	“0,0001” (кез-келген жолдан тұратын символ)

Суреттер объектісі.

Суреттер объектісі (**Image**) формаға *.bmp, *.emf, *.ico, *.wmf типті файлдардан графикалық объектіңою үшін қолданылады. Оның бұрынғы айтылған қасиеттерден басқа мынадай қасиеттері бар:

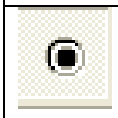
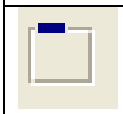
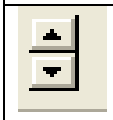
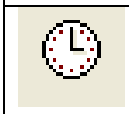
Кесте-6. Суреттер объектісі (**Image**) формасындағы файлдардың графикалық обьектісі

Қасиеттері	Қасиеттердің сипаттамасы	қабылданатын мәндерінің мысалы
Center	Орнатылған өріске қатысты центрі бойынша туралау	True, False
Picture	Графикалық файл атауы	Сұхбат терезесінде беріледі
Stretch	Берілген объекті өлшеміне сәйкес сурет өлшемін келтіру	True, False
AutoSize	Сурет өлшеміне сәйкес объект өлшемін келтіру	True, False

Delphi ортасының **Button,CheckBox, RadioButton, ScrollBar, TrackBar, Timer** компоненттерінің қасиеттерімен танысу және оларды қарапайым программалар құруда қолдану.

Кесте-7. Басқару элементтері:

Пиктограмма	Компонент	Орналасқан бет	Сипаттамасы
	Button (командалық батырма)	Standard	Командаларды іске асыру батырмаларын жасауда қолданылады

	BitBtn (суреті бар батырма)	Additional	Биттік графикасы бар батырмалар құруда қолданылады
	SpeedButton (Сурет пен фиксациясы бар батырма)	Additional	Жылдам батырмалар(командалық меню көшірмесінен тұратын инструменьтальдық панельдер жасау үшін қолданылады)
	RadioGroup (радиобатырмалар группасы)	Standard	Компонентте бірнеше радиобатырмалар орналастыруға болады, бірақ басқа басқару құралдарын орналастыруға болмайды.
	RadioButton (радиобатырма)	Standard	Қолданушыға көптеген альтернативалар ішінен біреуін таңдауға мүмкіндік береді
	GroupBox	Standard	Бір-бірімен байланысқан басқару элементтерінің (RadioButton, CheckBox т.б.) байланыстыру элементтері.
	UpDown (Батырма-счетчик)	Win32	Батырма – счетчик Edit және басқа компоненттермен қосылып, санды ақпаратты енгізуге қолданылады.
	CheckBox (Жалауы бар бақылаушы индикатор)	Standard	Қолданушыға операторларды іске қосуғажәне істен шығаруға көмектеседі.
	CheckListBox (Индикаторлар тізімі)	Additional	ListBox тізімінің CheckBox және индикаторларының қасиеттерінің 1 компоненттегі комбинациясы
	TrackBar (Сырғытпа)	Win32	Сырғытпа ретіндегі басқару элементі
	ScrollBar ()	Standard	Windows-ң стандартты айналдыру сызығының рөлін атқарады.
	Timer	System	Белгіленген уақыт интервалында процедуралар мен функцияларды іске қосу үшін қолданылады.

RadioButton – таңдау мүмкіндігін қамтамасыз ететін батырма.

Қасиет	Қасиет сипаттмасы	Қолданылатын мәндерінің

		мысалы
Caption	Батырма жанында пайда болатын мәтін	-
Alignment	Батырманың қай жағынан мәтін пайда болатынын көрсетеді	taLeftJustifi-сол жақ taRightJustifi -оң жақ
Checed	Қолданушының батырманы таңдаған, таңдамағандығын білдіреді	True False

CheckBox қолданушы қандай да бір опцияларды қосу, ажырату үщін немесе қалыпты индикациялау үшін қолданылады. Қолданушының әрбір шерткен кезінде оның жағдайы келесідей үш түрде өзгереді: айқындалу (қара жалауша пайда болуы), аралық (индикатордың қоңыр терезесі және қоңыр жалауша), айқындалмау (индикатордың бос терезесі). Осы үш жағдайға **State** қасиетінің 3 мәні сәйкес келеді: **cbChecked, cbGrayed, cbUnChecked.** Бірақ бұл үш жағдай **AllowGrayed** қасиетінің мәні **true** болғанда ғана мүмкін. Егер **false** болса, онда айқындалған және айқындалмаған болады.

TrackBar компоненті қолданушы жұмыс істеу істеу барысында тышқанмен және пернелермен жылжыта алатын сырғытпа ролын атқарады. Осылай қолданушы қандай да бір процестерді дыбысты көбейту, сурет өлшемін үлкейту және т.б. әрекеттерді орындай алады. Ол тік және көлденең орналаса алады.

Position сырғытпаның орналасу позициясын береді. Ол бүтін типті. Min – 10, Max – 10 - 0 мен 10 арасында өзгертуге болады. Orientation – сырғытпа бағытын береді: trHorizontal -тік, trVertical – көлденең.

TickMarks - компоненттің орналасуына байланысты шкаланың орналасуын береді.

tmBottomRight – төменгі сол жоғарғы жақ немесе жоғарғы оң жақ, компоненттің орналасуына байланысты.

tmTopLigth – жоғарғы сол жақ.

tmBoth – екі жағынан да.

TickStyle шкаласының суретінің пайда болу әрекеті:

tsNone - шкаланың жоқ болуы.

tsAuto - автоматты түрде салыну.

tsManual – Value өлшеміне сәйкес шкала белгісін SetTick (Value: Integer) әдісі арқылы программалық салу.

TickStyle = tsAuto кезінде шкала белгілерінің жиілігі Frequency қасиеті арқылы анықталады.

SelStart және SelEnd қолданушыға қандай да бір ақпарат беретін визуалды түрде шкаладан белгілі бір аралықты айқындау үшін

қолданылады, мысалы қолдануға болатын мәндер аралығы. Бірақ қолданушының одан шығып кетуіне ешқандай кедергі болмайды.

ScroolBar WINDOWS – тың стандартты айналдыру сызғышы ролін атқарады. Position қасиеті TrackBar компонентінің мәнімен бірдей. Kind қасиеті орналасу бағытын sbHorizontal, sbVertical береді. SmallChange және LargeChange қасиеттері батырманы шерткендегі «азың, «көбің жылжытуды білдіреді.

Қолданушының айналдыру сызғышының жылжытпасының орын ауыстыру оқиғасы OnScroll. Оған жүгіртпе орны – ScrollPos жылжытпаның орын ауыстыруының түрін беретін ScrollCode параметірі беріледі.Оның мәні:

Кесте-8 . SmallChange және LargeChange қасиеттері

Мән	Сипаты
scLineUp, scLineDown	«азың жылжыту: жоғары, төмен, оңға, солға
scPageUp scPageDown	«көбін жылжыту: жоғары, төмен, оңға, солға
scPosotion	Қолданушы жылжытпаның орнын ауыстырып, босатты
scTrack	Қолданушы жылжытпаның орнын ауыстырып жатыр
scTop,scBottom	Жоғары сол жақ немесе төменгі оң жақ бұрышқа жылжыды
scEndScroll	Жылжыту аяқталды

Timer қосымшада уақыт интервалын береді.Таймер мультипликацияны синхронизациялау, қолданушы көп уақыт жұмыс істемеген терезелерді жабады, экран сақтаушысын қосу, оқыту программаларында жауап уақытын беру т.б. жағдайларда қолданылады. Таймер форманың кез келген жеріне орналаса алатын визуалды емес компонент. **Interval** – милисекунд бойынша уақыт интервалы. **Enabled** – қол жеткізу мүмкіндігі. Егер **Interval =0** немесе **Enabled = False** болса, таймер жұмысын тоқтатады.

Символдық типтер.

Delphi-де екі символдық тип бар.

Типтер	диапазоны	байт
AnsiChar	0 - 255	2
WideChar	0 - 65535	2

Char – символдық типі AnsiChar типіне эквивалентті. AnsiChar типінің мәні Ansi форматының кеңейтілуімен сәйкес реттелген символдар жиынынан алынады. WideChar мәні халықаралық симводар жиыны Unicode-қа сәйкес реттелген символдар.

Ord(Ch)-символдық типтің бүтін мәнін береді, яғни Ansi кестесінде осы символдың реттік нөмері. Мұндағы Ch-символдық тип.
UpCase()-кіші регистрдегі әріптерді үлкен регистрдегі әріптерге ауыстырады.

Логикалық тип.

Типтер	байт
Boolean	1
Bytebool	1
Bool	2
WordBool	2
LongBool	4

Логикалық тип екі мәнді ғана қабылдайды. True (ақиқат) немесе false (жалған), осы екі мәннің бірін қабылдайды.

Саналатын тип.

Саналатын тип дегеніміз барлық мәндерінің жиыны тізім түрінде берілетін тип. Саналатын тип Type бөлімінде сипатталады.
Мысалы:
Type Ai=(KA,AK,NA,SA,MA,MM,MU,TA,KU,KZ,KP,GT);
Var x,y:Ai;
X,y айнымалылары осы саналатын 12 мәнді ғана қабылдайды.
Ord() - реттік нөмірін көрсетеді.
Pred() - алдындағы символдарды анықтайды.
Succ() - келесі символдарды анықтайды.

Аралық тип.
Аралық тип дегеніміз мәндері жиынының шекаралары арқылы берілетін тип.
Жазылу форматы:
Type
    <тип аты>=A..B;
Var
    <айнымалы аты>: <тип аты>;
Мысалы:
Type
    Kun=1..31;
Var
    Ai: Kun;

Ай-күн-уақыт типі.

тип	байт
TDateTime	8

Ағымдағы ай-күнді анықтайды.

Бір өлшемді массив

Массив – бұл бір атауға ие бір типтегі айнымалылар жиынтығынан тұратын мәліметтердің құрылымы. Массивтерді – табиғаты бір текті болып келетін ақпараттарды сақтау үшін қолдану ыңғайлы, мысалы кестелер және тізімдер.

Массив, программаның кез – келген басқа айнымалылары сияқты алдын – ала айнымалыларды сипаттау бөлімінде сипатталуы тиіс. Массивті сипаттау құрылымының жалпы түрі:

Аты: **array** [төменгі_ индекс..жоғарғы_индекс] **of** типі;

Массивтерді сипаттау мысалдары:

```
t: array [1..3] of real;
k: array [0..2] of integer;
n: array [1..30] of string[25];
```

Массивтің жоғарғы индексін тұрақтылар бөлімінде анықтау:

```
const
        N=18; // фамилиялар саны
        S=25  // фамилияның ұзындығы
var
        t: array [1..N] of string [S];
```

Массив элементін программада қолдану үшін массив атын және тік жақша ішіне номерін (индексін) көрсету керек.

```
t[1]: = 'Асанов ' ;
```

Әдетте массивтермен жұмыс істегенде мынадай амалдар орындалады:

1) Массивті шығару;
2) Массивті енгізу;
3) Массивтің ең кіші немесе ең үлкен элементін іздеу;
4) Массивті сортау.

Delphi програмистке графиканы: схемаларды, (чертеж) сызбаларды, иллюстрациялады шығаратын программаларды жасау мүмкіндігін береді.

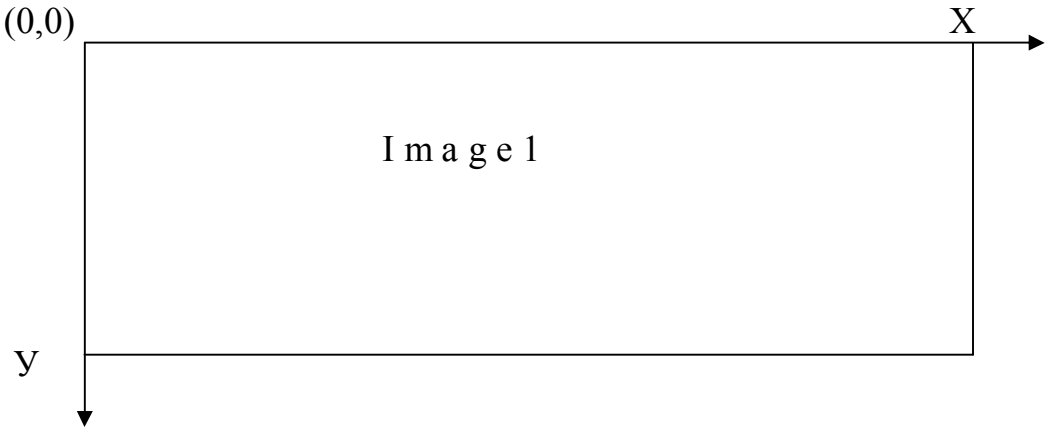
Программа графиканы обьектінің бетіне (формальды немесе Image компонентін) шығарады. Обьектінің бетіне *Canvas* қасиеті сәйкес келеді. Обьектінің бетіне графикалық элементті (түзу сызық, шеңбер, төртбұрыш

және т.б) шығару үшін, осы обьектінің *Canvas* қасиетіне сәйкесінше әдісті қолдану қажет. Мысалы, мына құрылым

Form1.Canvas.Rectangle (10, 10, 100, 100)

программа терезесінде төртбұрыш сызады. Сонымен *Canvas* Қасиеті – *TCanvas* типіндегі обьект. Бұл типтің әдістері графикалық (элементтердің) бейнелердің: түсін, сызық стилін және қалыңдығын, тұйық аймақты толтыру түрі мен түсін, текстік ақпаратты шығаруда шрифт сипаттамаларын орналастыру мүмкіндігін береді. (Canvas – бет, сурет салу үшін жазықтық)

Сурет салатын жазықтық жеке нүктелер – пикселдерден тұрады. Экран жазықтығындағы нүктелердің координаттары былай анықталады:



Жазықтық өлшемін *Hight* және *Width* қасиеттері арқылы алуға болады. Жазықтық бетінде графикалық бейнелерді салу үшін қалам мен қылқаламды (кисть) пайдаланамыз.

Қалам сызу үшін, ал қылқалам – бояу үшін қолданылады. Бұларға *Pen* (қалам) және *Brush* (қылқалам) қасиеттері сәйкес келеді. Бұл *Tpen* және *Tbrush* типіндегі обьектілерді, яғни бұл обьектілердің мәндері шығарылатын графикалық элементтердің түрін анықтайды.

Қалам сызатын сызықтың түрін *TPen* обьектісінің қасиеттері анықтайды. Олар:

Color – сызық түсі
Width – сызық қалыңдығы
Stule – сызық түрі
Mode – бейнелеу режимі

Color қасиеті қалам сызатын сызықтың түсін береді:
ClBlack – қара

ClGreen – Жасыл
Clolive – Сары (оливковый)
ClNavy – Тоқ көк (темно – синий)
ClPurple – Розовый
ClTeal – Жасыл – көк
ClGray – Сұр
ClSilber – Серебристый
ClRed – Қызыл
Cllime – Салатный
ClBlue – Көк
ClFuchsia – Ярко – розывый
ClWhite – Ақ

Width қасиеті сызықтың қалыңдығын пиксель бойынша береді. Мысалы, мына құрылым:

Canvas.Pen.Width: = 2

Сызықтың қалыңдығын 2 пиксель деп орналастырады.

Style қасиеті сызық түрін анықтайды: үзіліссіз немесе үзілісті штрих. Пунктирлі сызықтың қалыңдығы 1 – ден артық болмайды. Егер *Pn.Width* бірден артық болса онда пунктирлі сызық (үзіліссіз бір тұтас толық) сызық түрінде шығарылады. *Pen.Type* қасиетінің мәні сызық түрін береді.

PsSolid – біртұтас сызық
PsDash – пунктирлі, ұзын қысқа штрихтар
PsDot – пунктирлі, қысқа штрихтар
PsDashDot – пунктирлі, ұзын, қысқа штрихтар кезектесіп отырады. PsDashDotDot – пунктирлі, бір ұзын, екі қысқа штрихтар
PsClear – сызық көрінбей тұрады

Mode қасиеті сызық нүктелерінің түсі қалай бейнеленетіндігін анықтайды. Барлық сызық *Pen.Color* қасиетінің мәнімен анықталатын түспен сызылады.

Қылқалам (Canvas.Brush) тұйық аймақтарды сызуды және оған бояу құюды (бояуды) қамтамасыз етеді. Қылқалам обьект ретінде екі қасиетке ие.

Color – тұйық аймақты бояу түсі
Style – аймақты толтыру түрі, тилі

Color қасиетінің мәні ретінде *Tcolor* типіндегі мына кез – келген тұрақтыны пайдалануға болады:

- bsSolid – ***сурет*** толық бояу;
- bsClear – боямау;
- bsHorizontal – жатық сызықтар;
- bsVertical – тік сызықтар;
- bsFDiagonal – солға еңкіш тік сызықтар;
- bsBDiagonal – оңға еңкіш тік сызықтар;
- bsGross – торкөз;

- bsDiagGross – диагонал торкөздер.

2. Canvas обьектінің әдістері

1) Arc (x1, y1, x2, y2, x3, y3, x4, y4: integer) - әдісі доға салады.
2) Chord (x1, y1, x2, y2, x3, y3, x4, y4: Integer) әдісі эллипс сегментін салады.
3) Ellipse (x1, y1, x2, y2: Integer) – толтырылған эллипс салу.
4) Fill Rect (const Rect: TRect) – төтрбұрышты бояу.
5) FrameRect (const Rect: TRect) – боялмаған төртбұрыш.
6) LineTo (x, y: Integer) – көрсеткіш тұрған жерден бастап сызық сызу.
7) Pie (x1, y1, x2, y2, x3, y3, x4, y4: Integer) – эллипс секторын салу
8) Polygon (Const Points: array of TPoint) – толтырылған көпбұрыш
9) Polyline (Const Points: array of Tpoint) – боялмаған көпбұрыш.
10) Rectangle (x1, y1, x2, y2: Integer) – толтырылған төртбұрыш
11) RoudRect (x1, y1, x2, y2, x3, y3: Integer) – бұрыштары дөңгеленген боялған төртбұрыш салу.

Бұл әдістердің көмегімен сызық сызуда *Pen* – қалам қасиеті, ал ішкі аймақтарды толтыруда *Brush* – қылқалам қасиеті қолданылады.

12) MoveTo (x, y: Integer) - әдісі көрсеткіш – курсорды жаңа (x, y) орнына көшіру үшін қолданылады.
13) Draw(x, y: Integer; Graphic) – әдісі Graphic параметірімен берілген бейнені салу бетіне орналастырылады. Сурет тіктөртбұрышты аймаққа орналастырылады. Оның сол жақ бұрышы (x,y) нүктесіне орналастырылады.
14) StretchDraw(const Rect: TRect; Graphic: Tgraphic) – әдісі *Graphic* параметрімен берілген бейнені, *Rect* параметрімен анықталатын төртбұрышты аймаққа орналастырылады. Бұл кезде бейне бүкіл аймақты толтыру үшін сығылады немесе созылады.
15) TextOut (x, y: Integer; ConstText: String) – әдісі сурет салу бетіне текст бейнелеу үшін қолданылады. Оның (x,y) сол жақ жоғарғы бұрышы.
16) Graphic (Rect: TRect; x, y: Integer; const Text: String) – әдісі жоғарғы әдіс сияқты текстті бейнелеу үшін қолданылады, бірақ шығару аймағы *Rect* параметрімен берілген төртбұрыштың өлшемімен шектеледі.
17) TextHeight (Const Text: String) – әдісі Text жолы алып жатқан төртбұрышты аймақтың биіктігін береді.
18) TextWigth (const Text: String) – әдісі text жолы алып жатқан төртбұрышты аймақтың енін береді.
19) CopyRect(Dest: Trect; Canvas: Tcanvas; Source: Trect) – әдісі Source параметрімен берілген бастапқы орнынан сурет салу жазықтығының Dest төртбұрышты аймағына көшіру мүмкіндігін береді. Бұл кезде көшірілетін бейне Dest төртбұрышты аймағының өлшемдеріне сәйкес орналасады.

Delphi ортасының негізгі компоненттерімен танысу және олармен жұмыс істеуді үйрену.

Name (Атау) – формаға берілген атау. Ол Delphi объектілерінің негізгі қасиеттерінің бірі. Программаның жұмыс істеуі барысында Delphi объектіні осы атау бойынша ажыратып таниды. Delphi-дің формаға автоматты түрде алғашқы рет меншіктеген атауын (Form1) өзгертіп, басқа атау беруге болады. Ол үшін қасиеттер терезесінен Name қасиеті таңдалып, жаңа атау пернетақтадан енгізіледі.

Font (Шрифт) – формаға шығарылатын мәтін шрифтінің қасиеті. Оны таңдап, оң жағында көрінген көп нүкте (…) түймесін шерткен кезде Windows-тың сұхбат **Шрифті таңдау** терезесі көрінеді. Терезеден қажетті шрифт типін, өлшемін таңдап **ОК** түймесін шерту керек.

Caption (Тақырып) – форма терезесінің тақырыбына енгізілетін мәтін.

Color (Түс) – форманың түсін орнату қасиеті. Ол таңдалған кезде оң жағында тілсызық түймесі көрінеді. Тілсызық түймесі - қасиет мәнінің бірнеше екенінің белгісі. Тілсызық белгісін шерткен кезде мәндер (түстер терезесі ашылады. Тізімде көрінген қалаған түс таңдалған соң форма сәйкес түске боялады.)

Width (Ен), **Height** (биіктік) – пиксель, өлшем бірлігімен берілген форманың ені мен биіктігін орнату қасиеттері (бұл мәндер форманы қолдан кеңейту не сығу кезінде де автоматты түрде орнатылып қойылады).

Мысал 1 Delphi - де қарапайымдиалог құрға мысал келтірейік. Бірінші мысал қолданушы Диалог батырмасын басқан кезде "Бұл менің бірінші бағдарламам" сөзі шығуы керек.

1 . C: Progran Files: BorLand: Delphi7: Projects файлында өзіңізге жаңа папка құрыңыз, атын "Лаборатория№1" деп тағайындаңыз.

2.Delphi 7 программасын іске қосыңыз.

3.Формаға Standart бетінен Label1 және Button1компоненттрін орналастырып, қасиеттерін келесі кестедегідей етіп өзгертіңіз:

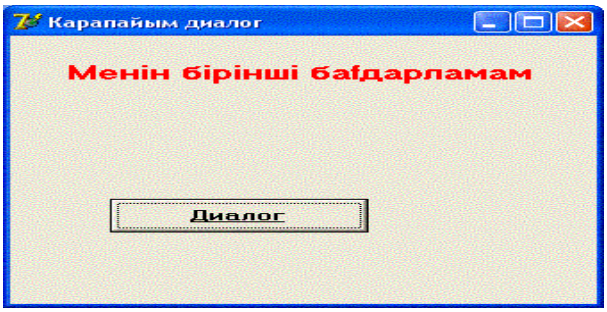
Label1 Caption:бос

Button1 Caption:"диалог"

4 .Button1 батырмасын екі рет басып, оның OnClick оқиғасына жауап ретінде

procedure Button1Click(Sender: TObject);

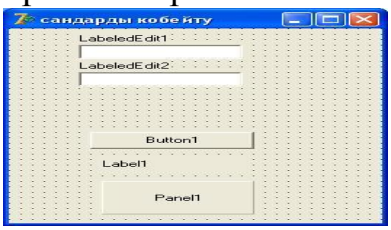
процедурасын жазамыз.



Қарапайым диалог құру

```
procedure Button1Click(Sender: TObject);
Begin
Label1.Caption:='Бұл менің бірінші бағдарламам';
end;
```

Мысал2. Delphi - де екі сннның көбейтіндісін есептейтін мысал келтірейік. Жаңа жоба ашып, Additional бетінен LabeledEdit - 2 компонентін, Standrt бетінен Panel, Button, Label компоненттерін алып, келесі суреттегідей етіп орналастырамыз.



Delphi - де LabeledEdit компоненті екі компоненттің орнына қолдануға болады, яғни Label және Edit компоненттерінің орнына.

LabeledEdit және Button компоненттерінің Caption қасиеттерін өзіңізге қолайлы, түсінікті аттарға ауыстырыңыз. Мысалы, "сан1", "сан2", "нәтиже", "ұзындық", "биіктік", "аудан", т.с.с., бұл сіздің сол сандар арқылы не есептегіңіз келетіне байланысты болады. Panel компонентінің бейнесін өзгерту үшін BevelInner және BevelOuter қасиеттерін тізімнен өзгертіп отырып өзіңізге қолайлысын таңдаңыз. Мысалы, **BevelInner=bvLowered**және **BevelOuter=bvRaised**

Button1 қасиетінің OnClick оқиғасына өңдеу процедурасын жазамыз.

```
procedure Button1Click(Sender: TObject); // нәтижені есептеу процедурасы
Begin
Panel1.Caption:=LabeledEdit1.Text+'*'+
LabeledEdit2.Text+'='+FloattoStr(StrtoFloat
(LabeledEdit1.Text)*StrtoFloat(LabeledEdit2.Text));
end;
```

Енгізу терезесі Delphi-дің стандартты **InputBox** функциясының терезесі. Программада InputBox функциясын пайдалану командасының жазылу үлгісі:

<айнымалы>:=InputBox(‘<тақырып>’,’<түсініктеме>’,’<мән>’);

мұндағы

айнымалы – мәні функция терезесіне енгізілетін жолдық типті айнымалы атауы (InputBox) функциясының мәні әркезде жолдық (String) типті. Мән меншіктелетін айнымалы (x) программада x: string түрінде сипатталуы тиіс;

тақырып – енгізу терезесінің тақырыбы ретінде жазылатын мәтін;

түсініктеме - енгізу терезесінің ішінде жазылатын түсініктеме мәтін;

мән – функция терезесі көрінген кезде оның енгізу өрісінде көрінетін мәтін. Әдетте оны бос етіп қалдырады. Мысалы, программада x:=4.7 меншіктеу командасын InputBox функциясын пайдаланып, мынадай түрде беруге болады:

x:=InputBox(‘Аргумент мәні‘,’x=’,’’);

Delphi-де нәтижені **ShowMessage** процедурасының терезесіне шығаруға болады. Процедураның жазылу үлгісі:

ShowMessage(s);

Мұндағы

s – жолдық типті өрнек. Егер ол сандық тпті болса, онда түрлендіру функцияларын қолдану керек.

Мысалы,ShowMessage(FloatToStr(s));

немесе ShowMessage(FloatToStrF(s,ffgeneral,7,3));

Мысал-1. Келесі программа форма бетіне күн-ай-жыл және сағатты шығарады. Бұл мысалды орындау үшін форма бетіне мынадай объектілерді орналастыру керек: белгілер (Label1, Label2, Label3, Label4,), кнопка (Button1).

Жұмыс барысы.

1. Delphi ортасын іске қосу.
2. Форма бетіне текстік белгілерді **Label1, Label2, Label3, Label4** орналастыр. **Label** қасиетінің мәндері төменгі кестеде келтіріледі.

Кесте-9. **Label** қасиеттері

Қасиеті	Мәні
Label1.Caption	Күн-ай-жыл
Label2.Caption	Уақыт
Label1.Font.Style	Bold
Label1.Font.Size	10
Label1.Font.Color	Purple
Label2.Font.Style	Bold
Label2.Font.Size	10
Label2.Font.Color	Purple
Label3.Caption	Бос ету
Label4.Caption	Бос ету
Label3.Font.Style	Bold
Label3.Font.Size	12

Label3.Font.Color	Red
Label4.Font.Style	Bold
Label4.Font.Size	12
Label4.Font.Color	Red
Button1.Caption	Күн-ай-жыл және сағатты көрсету

3. Сол сияқты формаға тағы да бірнеше текстік белгі орналастыр. Осы компоненттердің формада орналасуы мынадай:

4. Формаға орналастырылған Button1 кнопкасына екі рет шертіп, процедура денесін енгіземіз :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Label3.Caption:=DateToStr(Date);
Label4.Caption:=TimeToStr(Time);
end;
```

5. Программаны орындауға жіберіңіз. Ол үшін F9 пернесіне басамыз.

6. Жазылған программаны сақтаңыз. Ол үшін File → Save All командасын орындаймыз да, алдымен модуль атауын, одан кейін проект атауын береміз.

Мысал -2. Келесі программа берілген x-тің мәні бойынша функцияның мәнін есептейді. x=4.2 үшін $y=3x^2+5$ есептеу керек болсын. Біз екі түрлі жолды қарастырамыз:

1. Ең алдымен x-тің InputBox арқылы енгізіліп, ShowMessage арқылы шығарылады.
2. Одан кейін сол мән форма бетіндегі Edit компоненті арқылы енгізіліп, Label1 арқылы шығарылады. Бұл мысалды орындау үшін форма бетіне мынадай объектілерді орналастыру керек: енгізу компоненті (Edit1), белгі (Label1), кнопка (Button1).

1. Ең алдымен бірінші жағдайды қарастырайық.

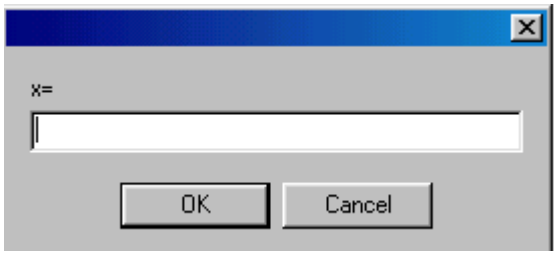
Жұмыс барысы**.**

1. Delphi ортасын іске қосу.
2. Форманы екі рет шерту. Сонда OnCreate оқиғасын өңдеуіш процедурасының дайындамасын көреміз.
3. Процедура денесін енгіземіз:

```
procedure TForm1.FormCreate(Sender: TObject);
var x,y: real;
    x1: string;
begin
x1:=InputBox(‘’,’x=’,’’);
x:=StrToFloat(x1); y:=3*x*x+5;
ShowMessage(FloatToStr(y);
```

end;

4. Программаны орындауға жіберіңіз. Ол үшін F9 пернесіне басамыз.



5. Терезеге х-тің мәнін яғни 4,2 санын енгізіп, ОК кнопкасын шертіңіз
6. Сонда төмендегідей нәтиже шығады:

2. Енді екінші жағдайды қарастырайық. Мұнда форма бетіне мынадай объектілерді орналастыру керек: енгізу объектісі (Edit1), белгі (Label1), кнопка (Button1).

Жұмыс барысы.

1. Delphi ортасын іске қосу.
2. Форма бетіне **Edit1, Label1, Button1** компоненттерін орналастыр. Олардың қасиетінің мәндері төменгі кестеде келтіріледі.

   Сол сияқты формаға тағы да бірнеше текстік белгі орналастыр. Осы компоненттердің формада орналасуы мынадай:
3. Формаға орналастырылған Button1 кнопкасына екі рет шертіп, процедура денесін енгіземіз :

```
procedure TForm1.Button1Click(Sender: TObject);
var x: string; y: real;
begin
x:=Edit1.Text; {х-тің мәнін енгізу компонентіне жазу }
y:= 3*Sqr(StrToFloat(x))+5;
Label1.Caption:='Функция мәні='+#13+FloatToStr(y); { #13 – келесі жолға өту коды}
end;
```

4. Программаны орындауға жіберіңіз. Ол үшін F9 пернесіне басамыз.
5. Х-тің мәнін енгізіп, іске қосу кнопкасына шертіңіз.
7. Жазылған программаны сақтаңыз. Ол үшін File → Save All командасын орындаймыз да, алдымен модуль атауын, одан кейін проект атауын береміз.

2.7. Delphi-7 ортасында есепті модельдеу.

Келесі физикалық проблемаларды тәжірибелік зерттеудің жоғарыдағы жүйелердің ортақ жақтары көп, сондықтан бірінші есепте виртуал зертханалық ортадағы жұмысты қарастырайық. Бұл ортаны жүзеге асыруды мүмкін түрі 2.7-суретте берілген. Басқару органдарының көпшілігі бас терезеде болғандықтан, зейінді негізінен оны қарастыруға бағыттаудың мәні бар. Келтірілген суреттен бас терезенің құрылысы жеткілікті оның көрінеді.

Қарастырылып отырған терезенің сол жағында 4 кіріс алаңы бар панель орналасқан оларды пайдаланып графикалық терезеде бейнелетін облысты реттеуге болады. Графикалық экранның схемалық тікбұрышының жанында X және У остерінің әрқайсысына ең кіші және ең үлкен мәндерін енгізу алаңдары қарастырылған. Бас терезеге енгізілетін координаталар және қалған басқа параметрлер тікелей "физикалық" бірліктерде көрсетіледі, сурет салу үшін экранның пикселдеріне айналдыруды бағдарлама өзіне алады.

Бас терезенің ортасында негізгі бөлігін алатын келесі панель есептің параметрлерін – VO серіктің ұшырудың жылдамдығын, dt уақыт қадамын (есептің қойылымында Δt) ; сол сияқты есептеуді қосу және тоқтатуды – беру үшін қолданылады. VO мәнін берудің екі жолы қарастырылған. Бірінші жағдайда график параметрдің бір мәні үшін тұрғызылады; ал "один график для" сөздерінен оңға қарай алаңда беріледі және бағдарлама "Счет-1" түймесімен санауға қосылады. Екінші жағдайда бірден бірнеше траектория автоматты есептеледі: санауға қажетті 0 шамасының бастапқы және соңғы мәндері кіргізіледі, тізімнен керекті траекториялары саны (олар 2-ден 11-ге дейін бола алады) таңдап алынады да "Счет-М" түймесі басылады. Есептеуді жүргізу, көрсету мен екі режимнің біріктірілуі кез-келген керекті траектория алып сандық тәжірибені өткізуге біршама ыңғайлы. Екі жағдайда да есептеуді кез-келген моментте "Стоп" түймесін басып тоқтатуға болады.

Барлық бағдарламалар үшін листинг қосымшада берілген.Бас терезенің оң жағында басқарудың үш пайдалы органы бар: болашақ траекторияларды суретінің түсін таңдаудың тізімін беретін және графикалық терезе мен кестелеудің мәнін тазалаудың екі түймесі (олар өздеріне салынған траекториялармен айырылады). Барлық түймелер бағдарлама жұмыс істегенде іске қосылады.

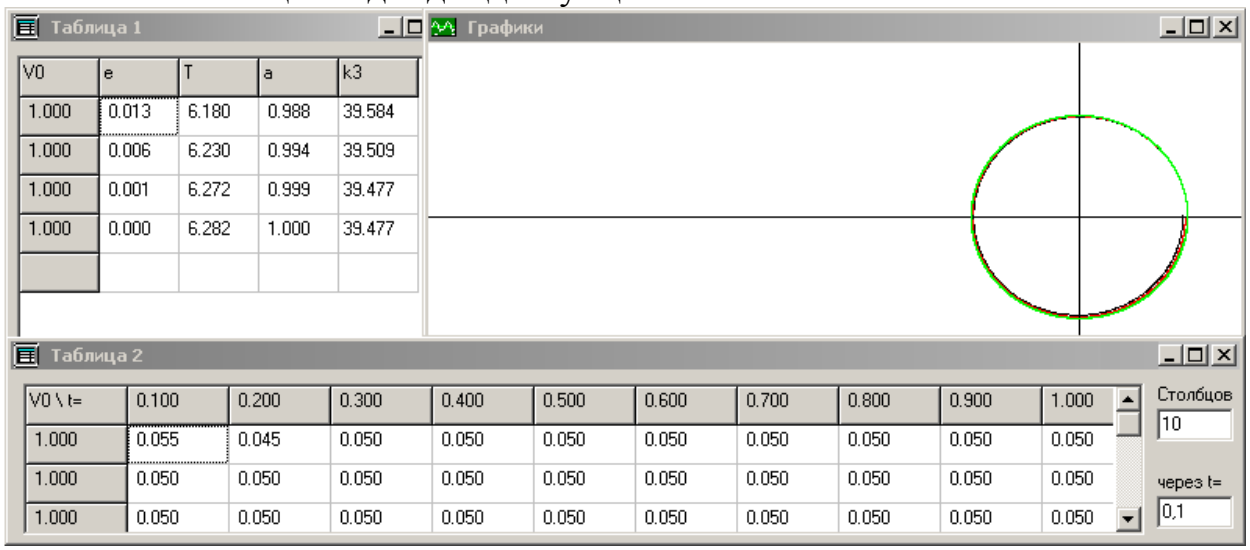
Сонымен, жоғарыда аталған басқару органдары серіктің физикалық моделін толық зерттеуді өткізуге мүмкіндік береді. Зерттеудің мақсатына сәйкесті оқытушының берген тапсырмасына сай параметрлерді таңдай ала отырып әртүрлі траекторияларды салып, өшіруге болады.

Әрбір трактория үшін функциялардың графиктері және интегралдық нәтижелері бар терезелерде ешқандай басқару органдары болмайды. Оның

есесіне, тұрғызу үшін берілген сурет-2.8 үшін параметрлерді енгізудің екі алаңы бар. Біріншісіне есептің физикалық мазмұнын туындайтын 2 кестенің белгілі бір сипаттамалары мәндері бекітілген траекторияның нүктелерінің санына тең бағандар саны енгізіледі. Екінші алаңға кестеге енгізілетін мәндер енгізілетін уақыт аралықтарның мәндері енгізіледі. Мысалы, суретте көрсетілген әрбір траектория үшін кестеге уақыттың 0.1, 0.2, ...1 мәндері үшін 10 нүктеден келеді.
"Итоговый график" терезесінде бірінші кестенің екі бағаны бойынша "көпіршіктермен" функция графигін тұрғызуға болады. Бағандардың номерлері бағдарламаның негізгі модулінде көрсетілген. Графикті тұрғызу "Построить" түймесімен жүзеге асады. Осы терезенің екінші түймесімен "көпіршіктердің" өлшемдері реттеледі.

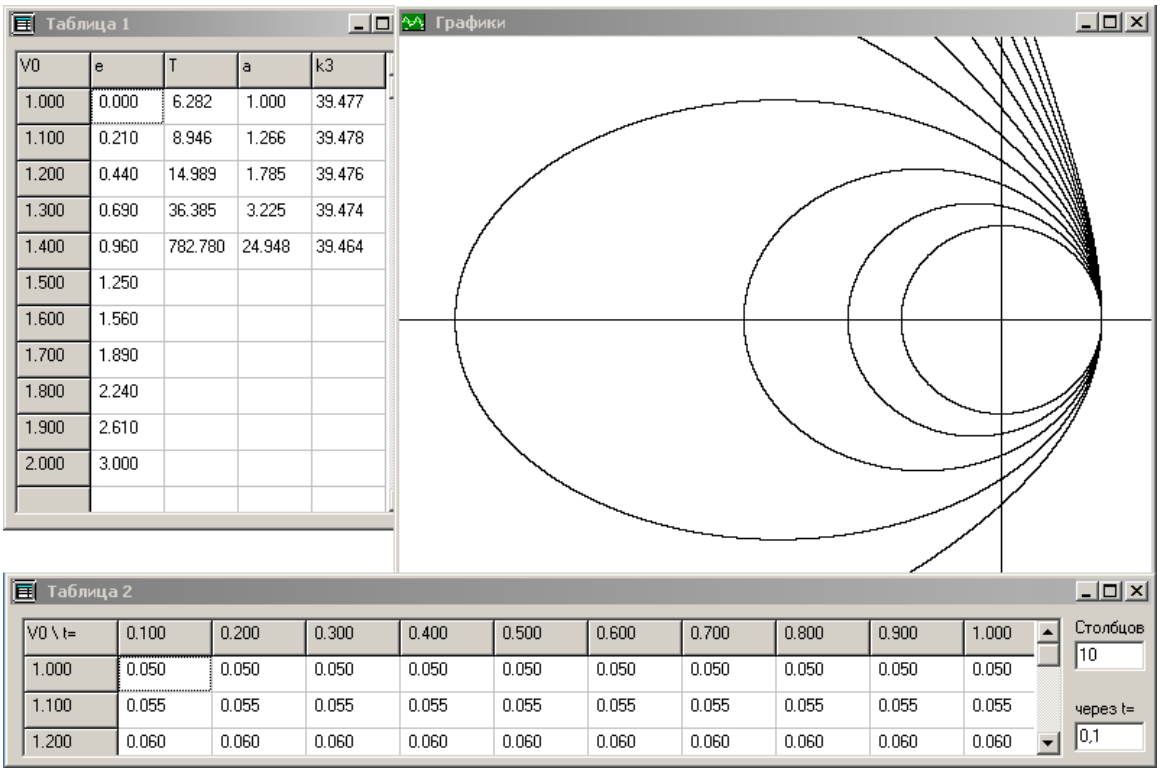
Серіктің қозғалысын тәжірибеде зерттеуді өткізейік. Сандық тәжірибені жүзеге асыру үшін алдымен траектория тұрғызудың dt уақыт қадамын таңдап алу керек. Тәжірибенің мазмұны бойынша осы интервалда үдеудің өзгерісін, жаңа координаталарды есептеу үшін жылдамдық өзгерісін ескермеуге болатындай жеткілікті аз шама болуы керек. Қадам ең аз да болмауы керек, бұл жағдайда траекторияны тұрғызу үшін көп уақыт керек болады. Масштаб ретінде жүйенің сипаттаушы параметрлерді алғандықтан, жүйенің өмір сүруінің уақыттық интервалы 1-ге тең болады. Сондықтан dt интервалы бірден әлдеқайда кіші болуы керек, мысалы, 0,01 тең деп таңдауға болады. Қажетті екінші параметр бастапқы жылдамдық. Есептеу есептің меншікті масштабында жүргізілгендіктен, бірінші тәжірибе үшін VO=1 болуы табиғи нәрсе. "Счет-1" түймесін басып, бір траекторияны тұрғызуға болады. Алынған траекторияны қарастырайық. Ол шеңберге ұқсас, бірақ ол серік тарту центрінен бір айналым жасағанда тұйықталмайды (2.8 -суреттегі графиктердің ішкі қисығы). Бұл санаудың дәлдігінің дәлдігі жеткіліксіз болуының салдары. Енді dt интервалын екі есе азайтайық: dt=0,005 және қисықтың түсін өзгертеміз. Әртүрлі dt үшін алынған нәтижелерді салыстыру үшін экран мен кестелерді тазаламаймыз. "Счет-1" түймесін басамыз. Нәтиже жақсарады, алайда, әлі де кішірек қадам керек. dt =0,001 деп алайық. Тұйықталған траектория алынады. Енді біз алынған қисықтардың интегралдық сипаттамаларын қарайық, есептеу дәлдігі артқан сайын γ эксцентритет кемиді және нольге жақындағанын көреміз. Мүмкін ноль болмас тек есептеу жеткіліксіз болар d=0,0001 дейін азайтайық. Жаңа түс және "Счет-1" түймесі (2.8 суретте есептеулер жасалғаннан кейінгі терезе күйі көрсетілген). Енді нәтиже жақсы! Эксцентритет нольге тең, үлкен жарты ось бірге тең, яғни траектория шеңбер. Қабылданған бірліктерде бірінші космостық жылдамдық 1, ал айналу периоды - 2π. Төменгі кестеден осы траектория үшін Кеплердің
2 –заңы орындалатыны көрініп тұр: бірдей уақыт аралығында радиус-вектор траекторияның әртүрлі бөлігінде бірдей аудан "сызады".

Енді серіктің басқа қандай траекториялары болатындығын қарастырайық. Бастапқы жылдамдықтар 1-ден 2-ге дейінгі аралықтарында болатын 11 сызықтан тұратын траекториялар сериясын қарастырайық. Алаң мен кестені тазалап, керекті параметрлер жиынын теріп "Счет-М" түймесімен бағдарламаны қосамыз. 2.8- суретте осы тәжірибенің нәтижелері көрсетілген. Бастапқы жылдамдық өскен сайын, эллипс X осінің бойында созылатыны көрінеді. Тарту центрі (координаталар басы) эллипстің бас фокусында орналасқан. Соңғы тұйық траектория VO=1,4 үшін (оның созылуы үлкен болғандықтан, экранға симайды) алынған. Бұл траекторияны эксцентритті 1-ге тең, оған әйкес бастапқы жылдамдық екінші космостық жылдамдыққа жуық.



2.8 сурет dt таңдаған кейінгі тәжірибеден кейінгі тәжірибелер терезесінің күйі.

Тәжірибенің соңында "Итоговый график" терезесінде серіктің айналу периодының оның бастапқы жылдамдығына тәуелділігін тұрғызамыз. Бұл қисық 2.9 суретте берілген.

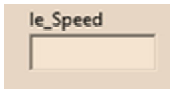


2.9 сурет. Траекториялардың сериясын есептегеннен кейінгі нәтижелер терезесінің күйі.

Нақты ортада көкжиекке бұрыш жасай лақтырылған дененің қозғалысы.

Нақты ортада көкжиекке бұрыш жасай лақтырылған дененің қозғалысын модельдейтін бағдарламаның сыртқы көрінісі 2.10 суретте берілген. Қарастырылып отырған терезенің сол жағында кірістің 4 алаңы бар панель орналасқан, оларды пайдалана отырып сәйкесті бастапқы шамалардың параметрлерін енгізуге болады.

Бастапқы жылдамдықтың υ_0 көкжиекке көлбеулік бұрышы - 

Бастапқы жылдамдық модулі [υ_0]- 

Ортаның кедергі коэффициенті k - 

Масса m - 

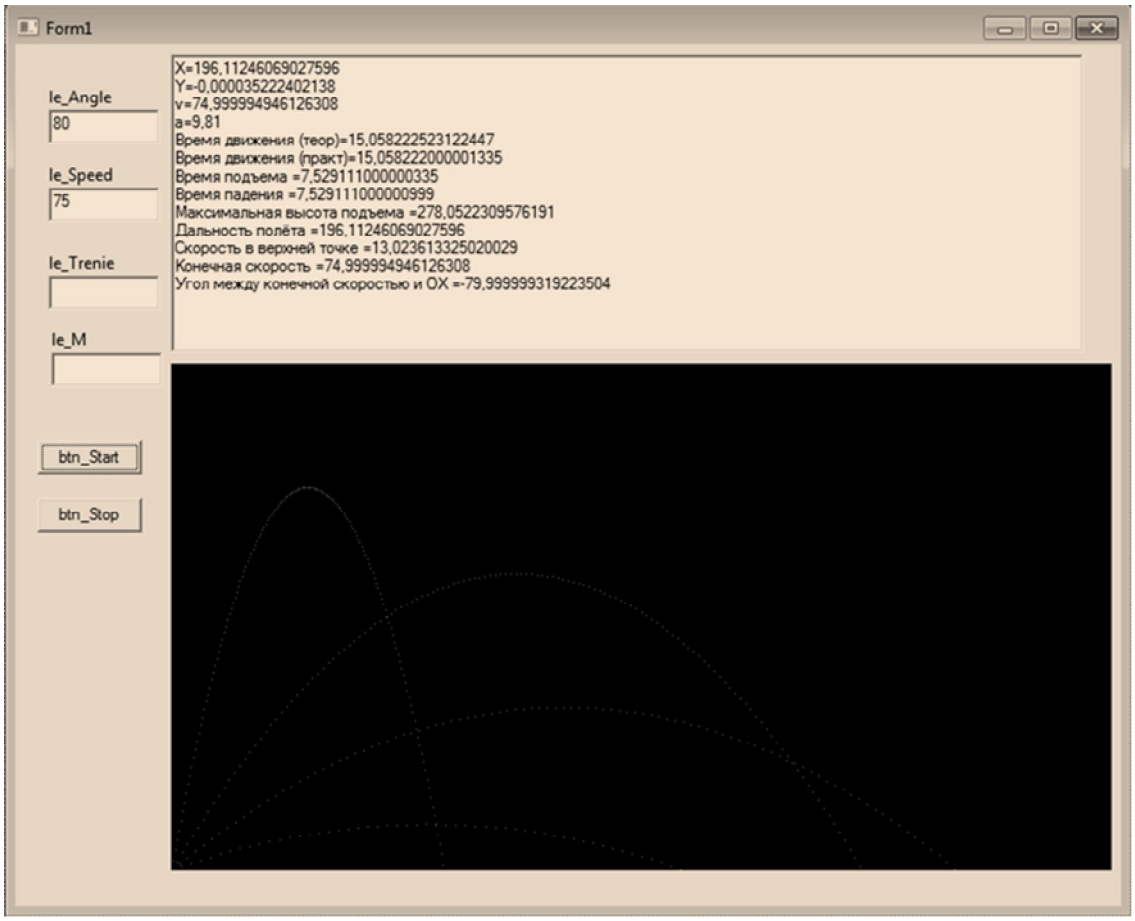
"Старт" түймесі – 

"Стоп" түймесі - 

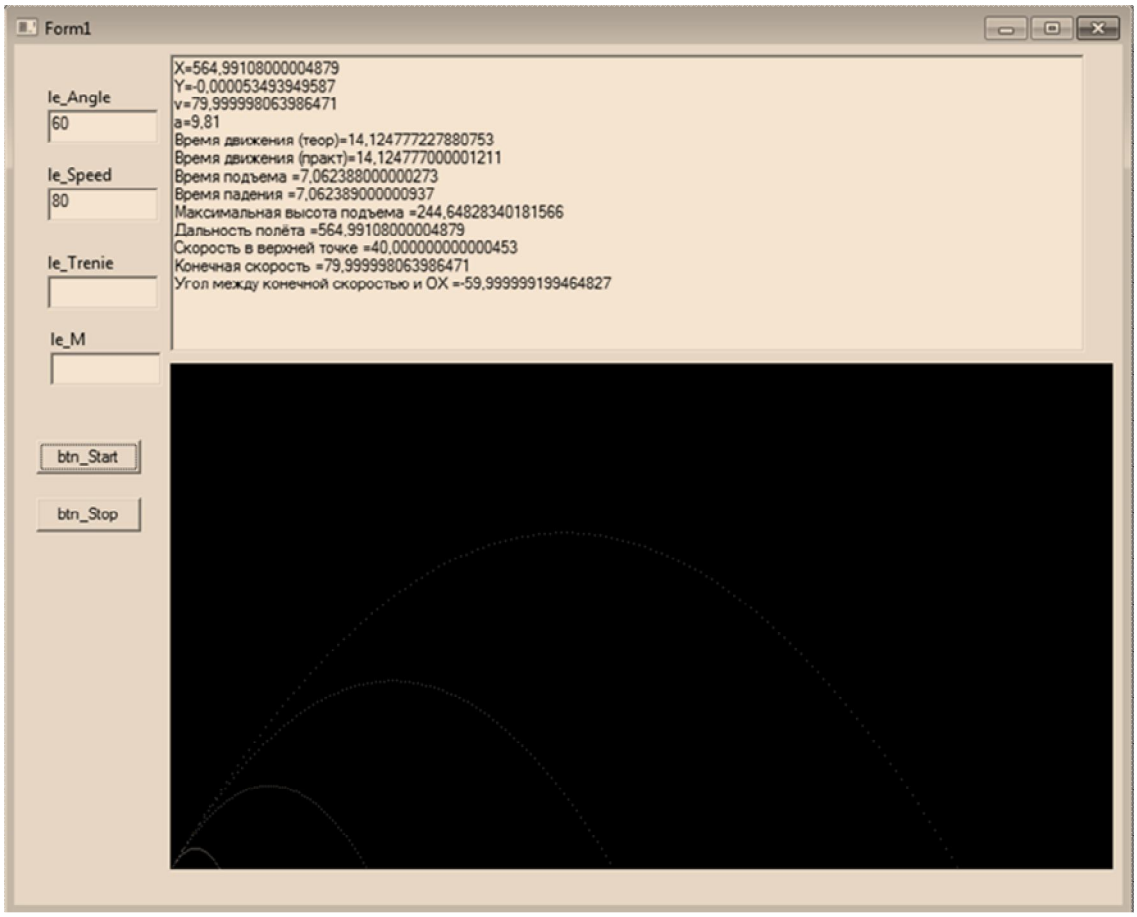
Бас терезеге енгізілетін параметрлерде, экранда көрінетін соңғы нәтиже көрсетіледі, 2.9-сурет салуға қажетті экран пиксельдеріне аударуды бағдарлама өзіне алады. Сандық есептеу бас терезенің негізгі жоғары жағынан орын алған Memol өріне шығарылады. Бұл текстi соңынан текст редакторына кірістіру үшін тікелей көшіріп алуға болады.

2.10 сурет. Бағдарламаны қосу алдында терезенің бастапқы көрінісі.

Бастапқы параметрлерді енгізгеннен кейін "btu start" түймесін басу арқылы қозғалыстың физикалық сипаттамаларын есептеу және траекторияның графигін тұрғызу автоматты түрде жүреді. Бұл есептеуді кез-келген уақыт моментінде ағымдардағы шамалардың мәнін сақтай отырып, "btu stop" түймесін басу арқылы тоқтатуға болады.



2.11 сурет Дененің υ=75м/с, α=20º, 40º, 60º, 80º мәндеріндегі қозғалысы



2.12 сурет Дененің α=60º, υ=20,40,60,80 м/с кезіндегі қозғалысы

Бұл жұмысты траекторияның әртүрлерін және оларға сәйкес физикалық сипаттамалардың есептің бастапқы шарттарына сәйкес тәуелділігін зерттеуге болады, мыс 2.11 және 2.12 суреттерде, сол сияқты кедергі күшін есептегенде 2.13 сурет.

2.12 -суреттегі есептің бастапқы параметрлері: α=60º, υ=80 м/с кезінде шешуін қарастырайық.

X=564,99108000004879
Y=-0,000053493949587
v=79,999998063986471
a=9,81
Қозғалыс уақыты (теориялық)=14,124777227880753
Қозғалыс уақыты (практикады)=14,12
Көтерілу уақыты = 7,062388000000273
Құлау уақыты = 7,062388000000937
Көтерілудің максимал биіктігі = 244,64828340181566
Ұшу аралығы = 564,99108000004879
Жоғары нүктедегі жылдамдығы = 40,000000000000457
Соңғы жылдамдығы = 79,999998063986471
Соңғы жылдамдық пен OX арасындағы бұрыш = -59,999999199464827

Көтерілу және құлау уақыты, қозғалыстың толық уақыты (теориялық және практикалық) мәндері жеткілікті дәрежеде сәйкес келеді. Тексеру және сәйкестік мақсатында жүргізілген салыстырмалы талдау компьютерлік модельдеу нәтижелері қолмен қағазда есептеу нәтижесімен жақсы қабысады).2.3- суреттегі параметрлерге ортаның кедергі коэффициент k қоссақ, күштің жылдамдыққа пропорционалдығын және қозғалысты тежейтіндігін ескерсек, төмендегі нәтижелерді аламыз.

$\alpha = 60^0$, v= 80 м/с и k=0,1

X=277,58961458098218

Y=-0,000029632417606

v=48,448673360061932

a=5,266558105561576

қозғалыстың уақыты (теор)- 14,124777227880753

қозғалыстың уақыты (прак)- 11,840855000000908

көтерілу уақыты = 7,062388000000273

құлау уақыты =4,778467000000634

көтерілудің максимал биіктігі =167,208330021615

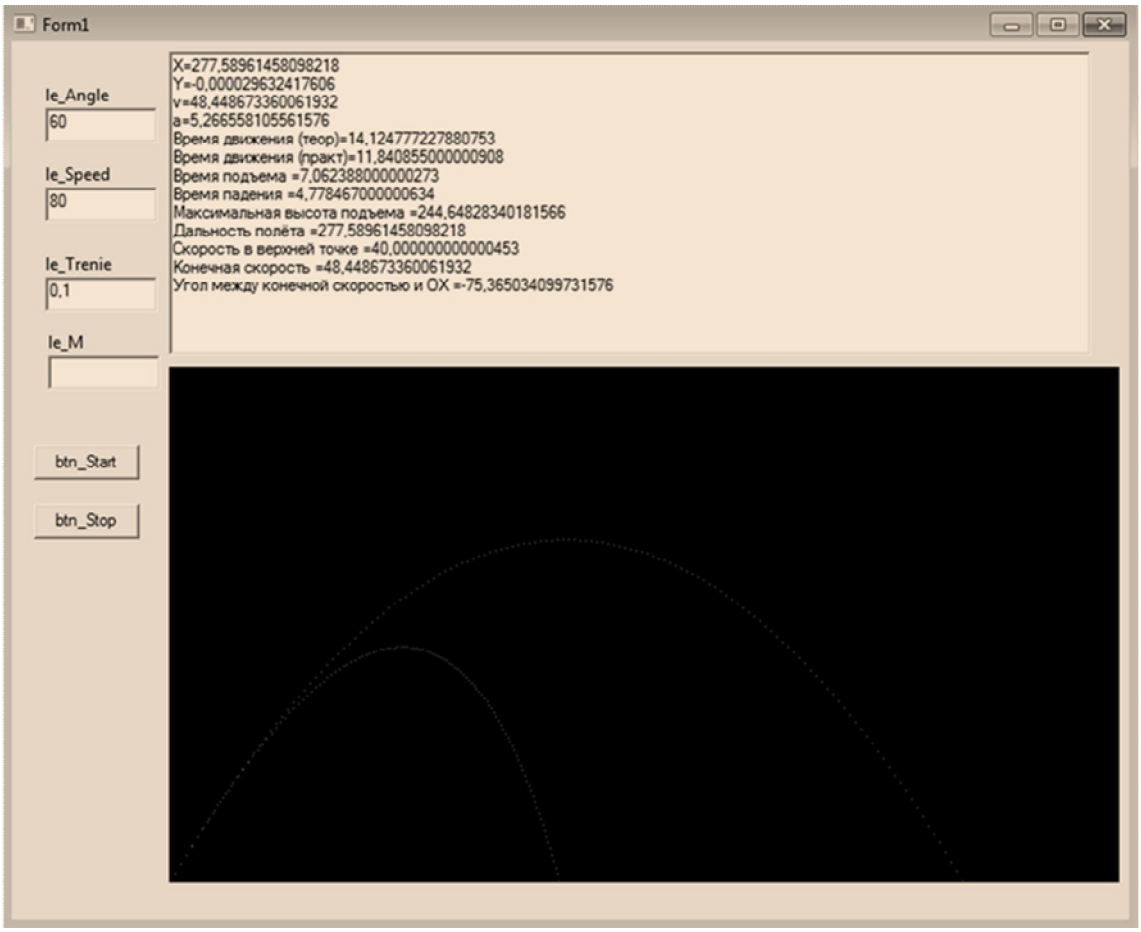
ұшу аралығы =277,58961458098218

жоғары нүктедегі жылдамдығы =25,349031410902118

соңғы жылдамдық =48,448673360061932

соңғы жылдамдық пен OX арасындағы бұрыш =-75,365034099731576

Мұнда, біз көптеген параметрлерді сән мәндері кемігенін, ұшудың максималь мәні теориялық мәнінен 287,40146541906661 қысқарады. Құлау уақыты көтерілу уақытынан кем.



2.13 сурет Дененің үйкеліс күші болмаған және болған жағдайдағы траекториялары

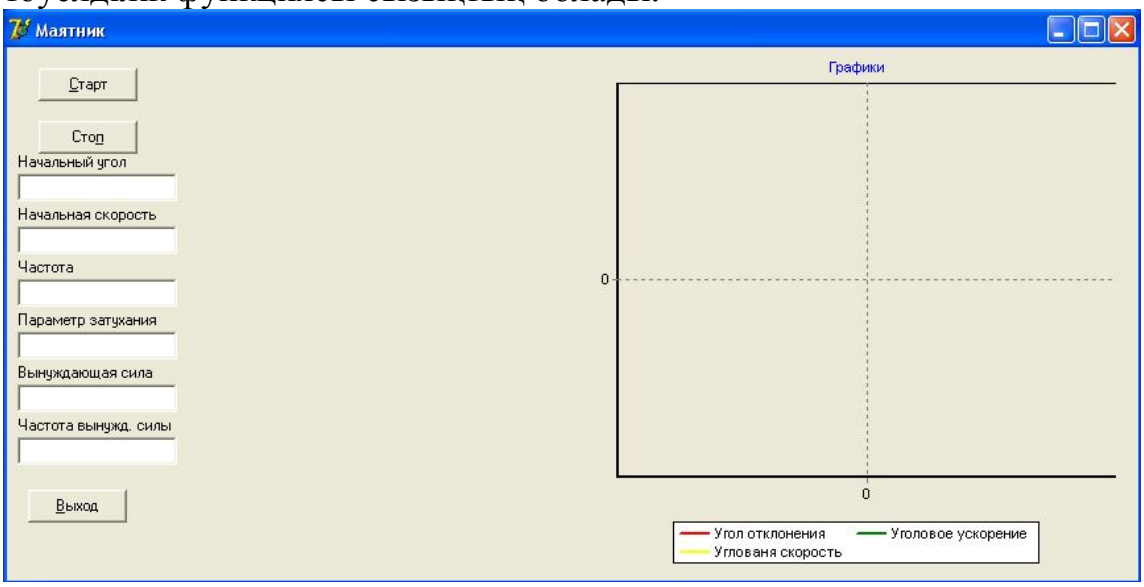
Физикалық маятник.

Есептің шешуін модельдейтін программаның сыртқы көрінісі 2.14 суретте келтірілген. Көрсетілген терезенің сол жағында 6 кіріс алаңы бар панель орналасқан, олар арқылы көрсетілген шамалардың параметрлерін беруге болады, "Старт", "Стоп" және "Выход" түймелері бар. Терезенің орта бөлігінде маятниктің тербеліс кезіндегі қозғалысын имитациялайтын динамикалық модель орналасқан (2.15 сурет). әрі қарай оң жақта ауытқу бұрышының, бұрыштық жылдамдықтың және бұрыштық үдеудің уақытқа тәуелділігі бейнелетін координаталар жүйесі орналасқан.

Бастапқы параметрлерді енгізгеннен кейін "Старт" түймесін басу арқылы қозғалыстың физикалық сипаттамаларының уақыттан тәуелділігі графигін автоматты тұрғызу қосылады. Берілген есепті "Стоп" түймесін басып, барлық ағымдағы шамалардың мәнін сақтай отырып, тоқтатуға болады.

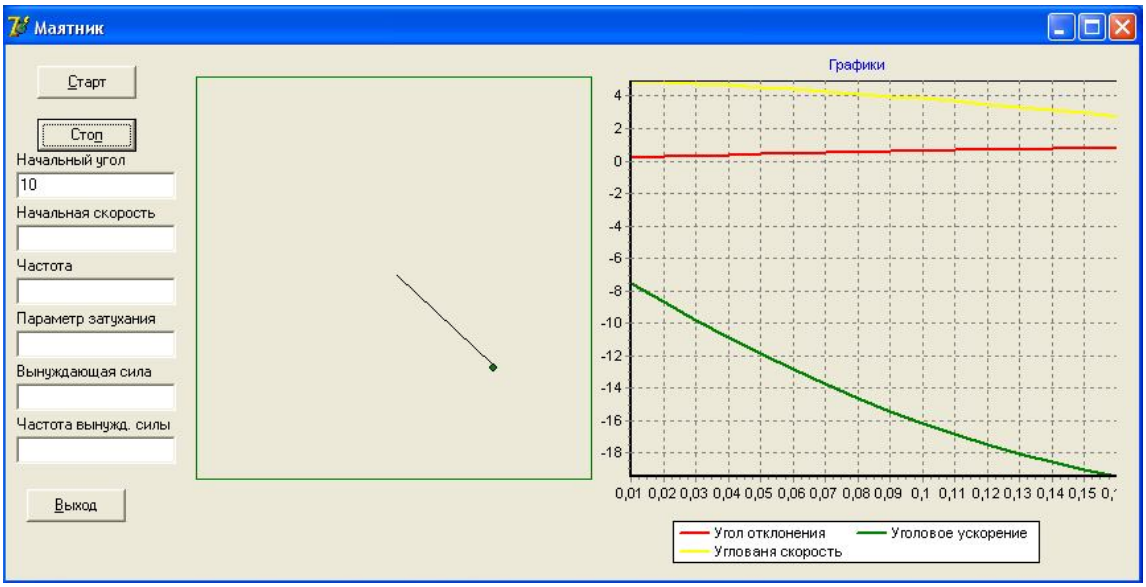
Тәжірибе нақты уақыт режимінде жүргізілгендіктен, тәжірибенің ұзақтығы өскен сайын графиктің бір бөлігінің мәні үнемі өсе береді (2.15-3.10 суреттер). Бастапқы уақыт моментінде оның мәні 0,0025 сек (3.4- сурет).

2.16 және 2.17- суреттердегі графиктер маятниктің тербеліс теңдеуі сызықтық емес екенін көрсетеді. Бұл "стержень-жүк" жүйесінің аса күрделілігімен байланысты, атап айтқанда: жүктің үдеуі Гук заңындағыдай, координатаға пропорционал емес, тепе-теңдік жағдайынан ауытқуды
(α бұрышының) күрделі функциясы бодып келеді. Егер бұл ауытқулар кішкене шама болса, sinα ≈α онда кішкене тербелістер моделінде тәуелділік функциясы сызықтық болады.

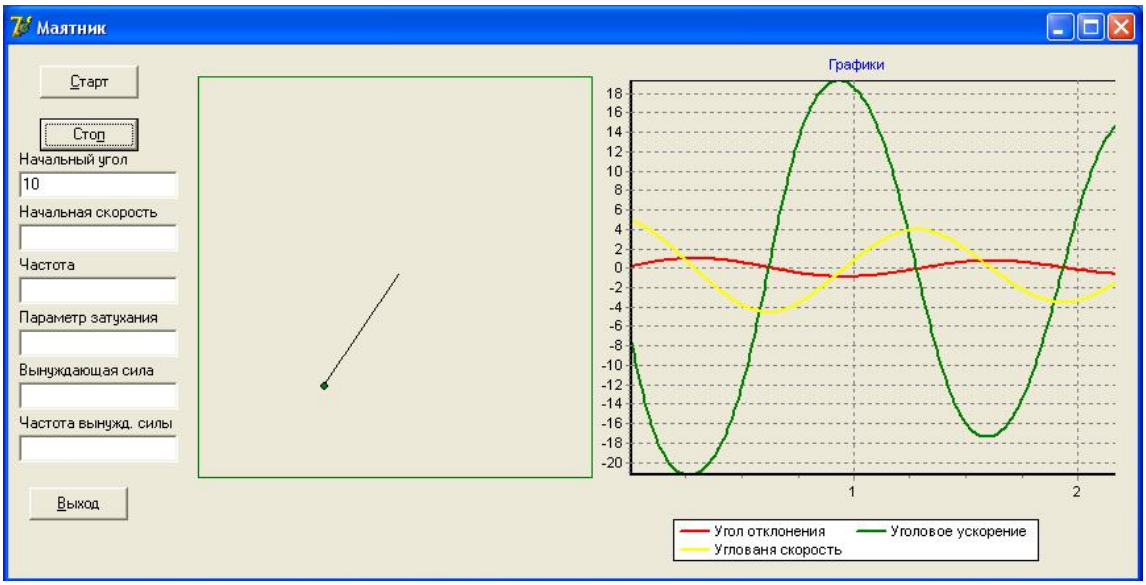


2.14 сурет Бағдарламаның қосу алдындағы терезенің бастапқы көрінісі.

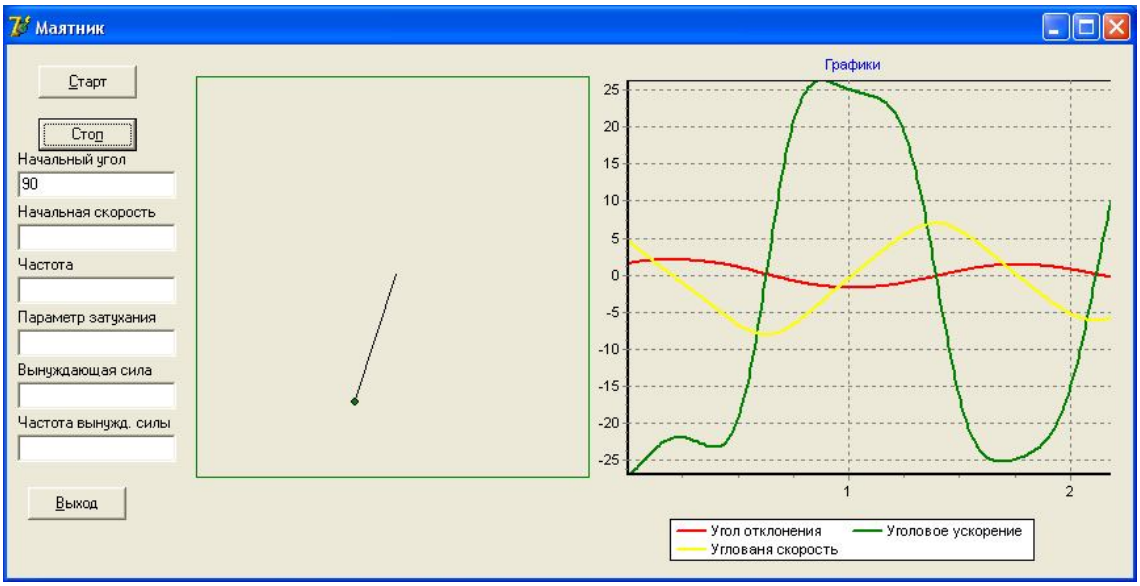
2.19 және 2.20 -суреттері жүйенің моделін шарнирдегі үйкелісті ескеріп және ескермей көрсетеді. Бұнда өшу коэффициенті тежеуші күштің өлшемі ретінде алынады. Екінші жағдайда тербелістердің өшуі біршама жылдам болады.
2.21- суретте физикалық маятниктің тербелісі мәжбүрлеуші күшті ескере отырып бейнеленген. Сыртқы күш белгілі бір бұрыштық жиілікті гармоникалық шама ретінде қарастырады. Графиктен бірінші резонанс 3,125-3,7 сек аралығында, екінші 10-11,25 сек аралығында болатындығыкөрінеді, аздап өсу 15-20 сек аралығында болады, әрі қарай тербелістер гармоникалық сипат алып орнығады.
2.22 -сурет қызғылықты нәтижені бейнелейді. Мұнда маятник өшпелі тербелістерге өту алдында қозғалмайтын абсолют қатаң шарнирдің айналасында 5 айналым жасайды.



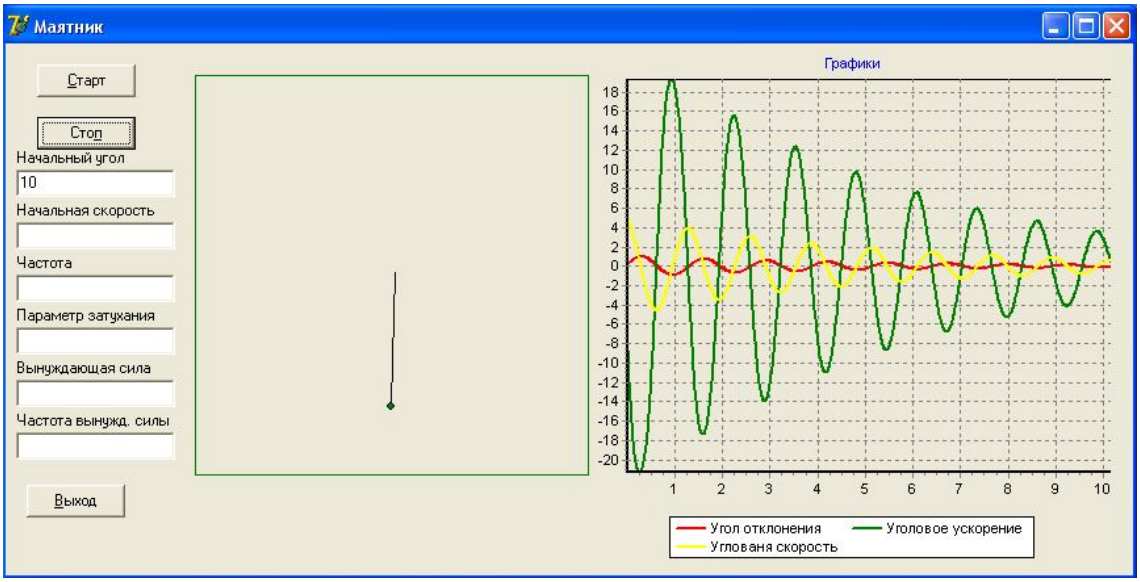
2.15 сурет. Бір бөлікке 0,0025 сек масштабында физикалық шамалардың тәуелділік графигі



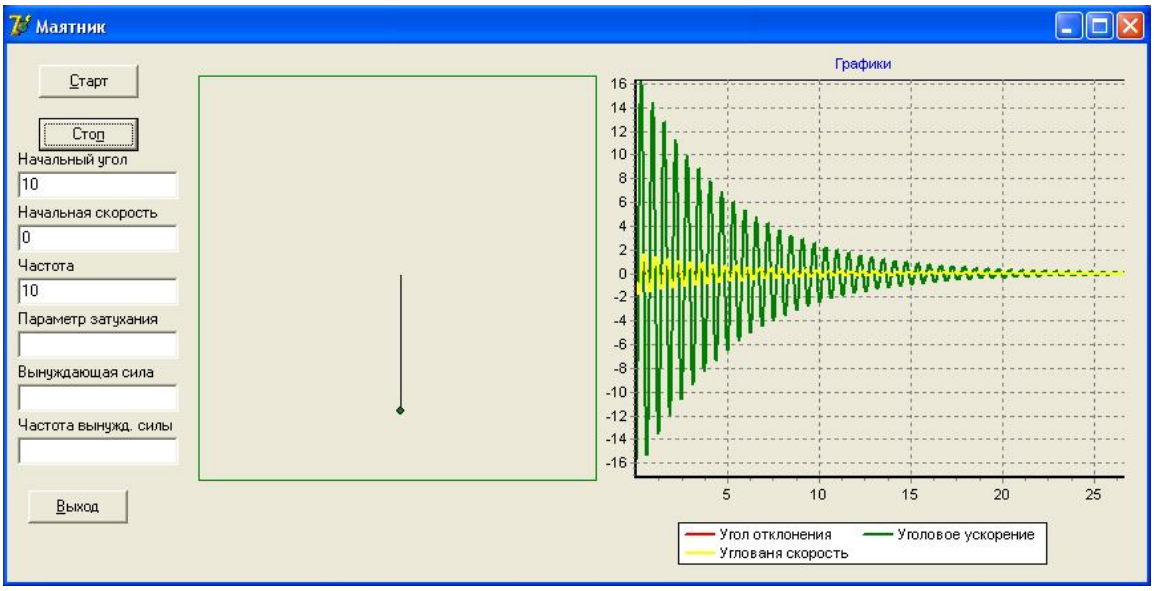
2.16 -сурет. α=10º кезіндегі физикалық шамалардың сызықтық тәуелділігі графигі



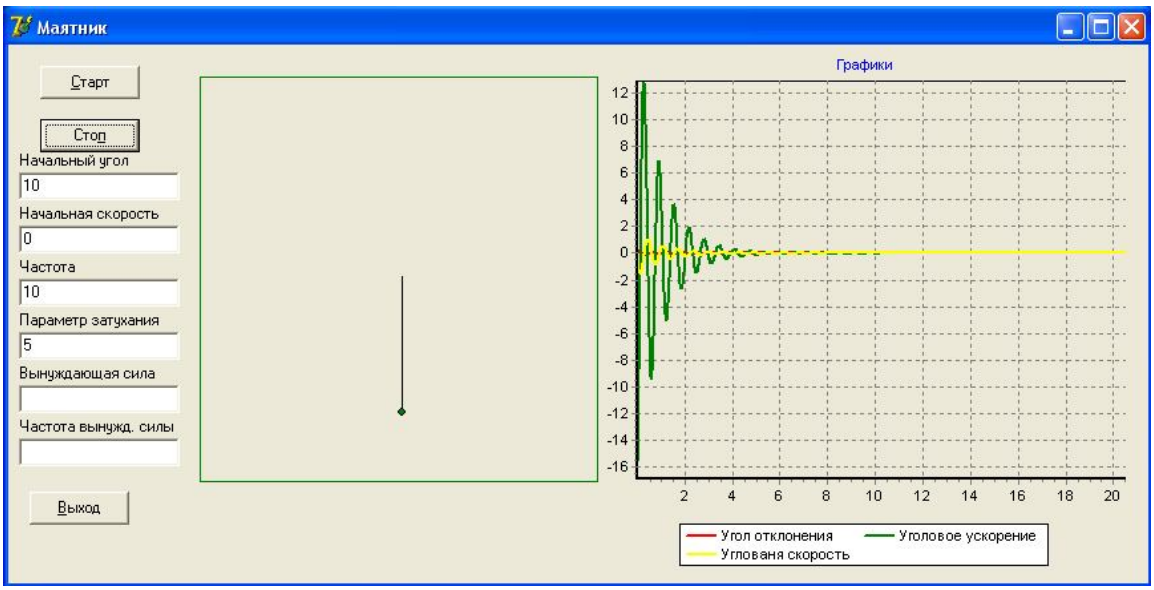
2.17- сурет. α=90º кезінде физикалық шамалардың сызықтық емес тәуелділігі графигі.



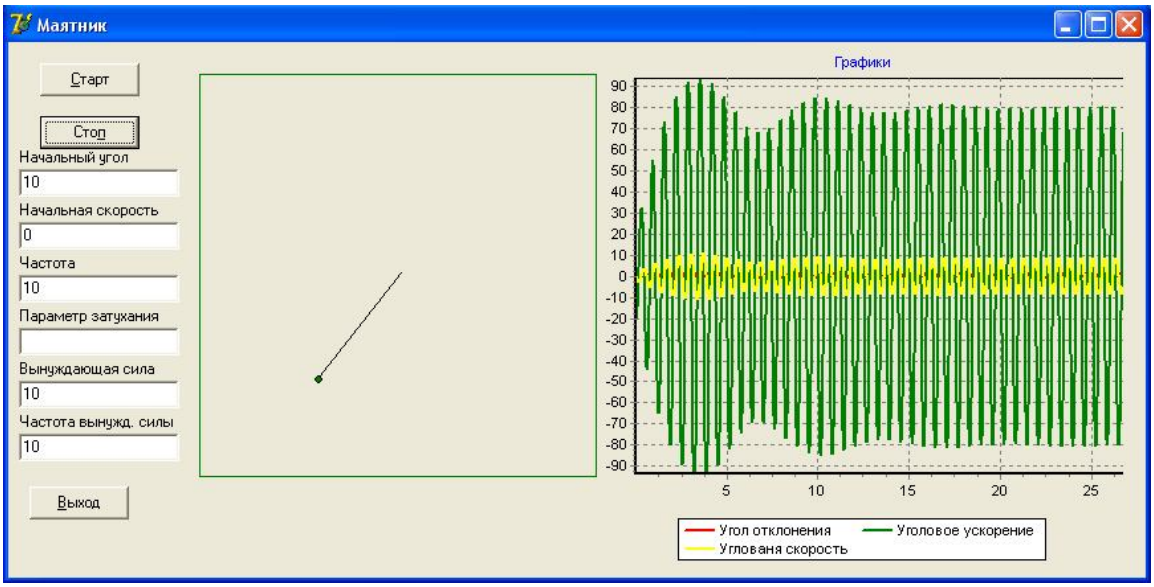
2.18 сурет. Бір бөлікке 0,25 сек масштабындағы физикалық шамалардың тәуелділік графигі



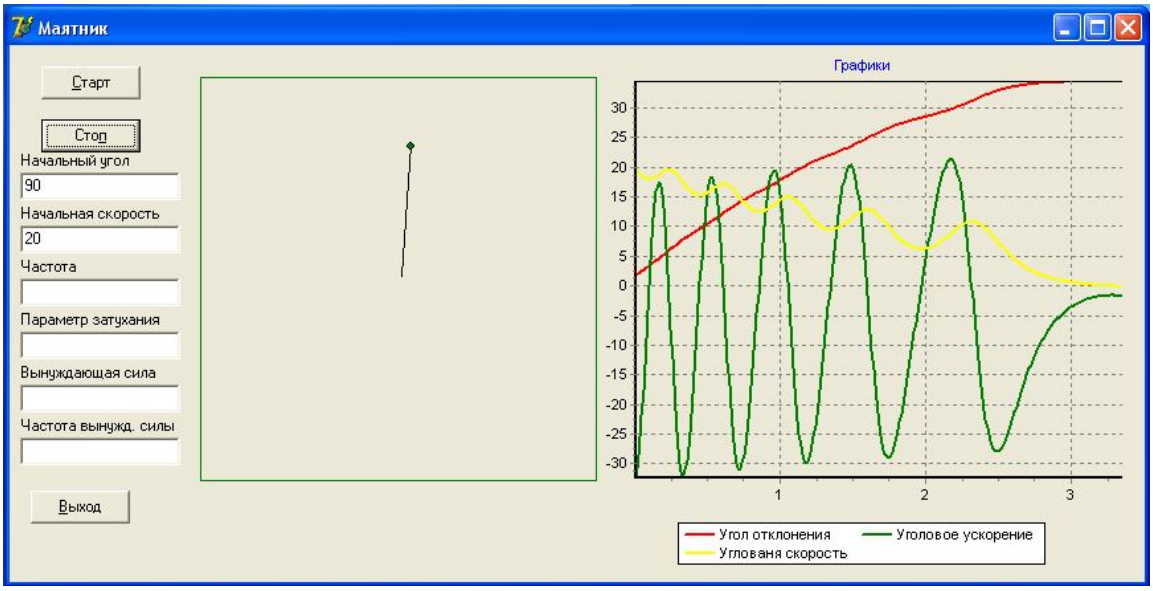
2.19 сурет. Физикалық шамалардың өшу параметрінсіз тәуелділік графигі



2.20 сурет. Физикалық шамалардың өшу параметрі мен тәуелділік графигі



2.21 сурет. Мәжбүрлеуші күшті ескерген кездегі физикалық маятниктің тербелісі

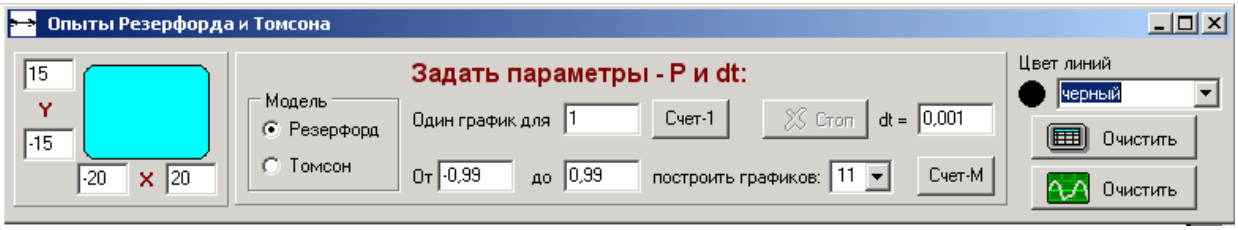


2.22 сурет. Айналатын физикалық маятник.

Атом моделі

Ұсынылып отырған виртуал ортада Томсон және Резерфорд модельдері бір бағдарламаға біріктірілген. Басқару органдары орналасқан негізгі терезе көрінісі 2.23 суретте көрсетілген. Барлық түймелер мен терезелер 1 есептегідей функциялар атқарады.Қосымша модельдер терезесі қосылған.

Серпімді шашырауды тәжірибеде зерттеуді өткіземіз. Модельдер терезесінен Томсон моделін таңдап аламыз. Санау тәжірибесін жасауда ең алдымен траекторияны тұрғызу үшін dt уақыт қадамын таңдап алу қажет.



2.23 сурет. Бөлшектердің шашырауы есебі шешу басқару терезесінің көрінісі.

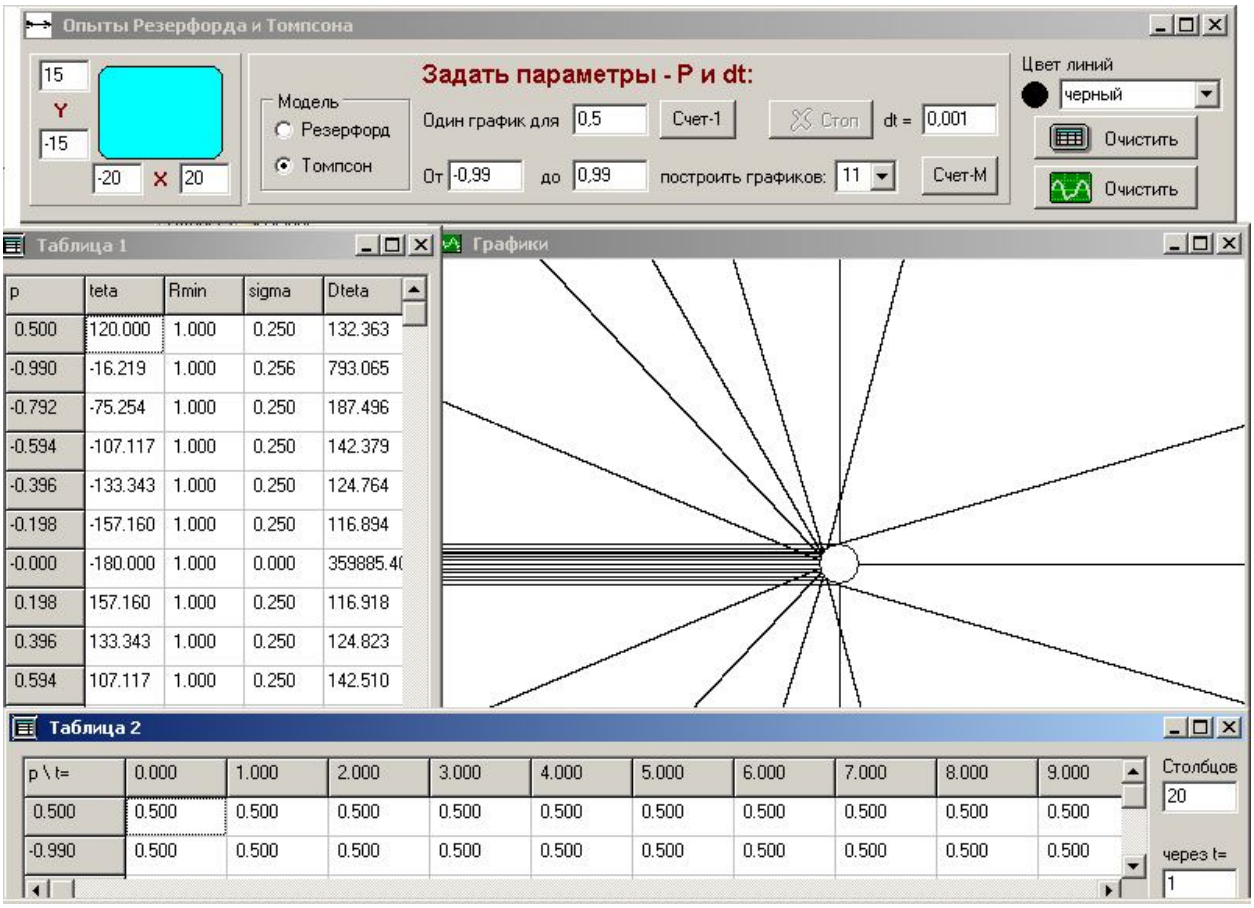
Траекторияның түзу сызықты бөлігінде бірқалыпты қозғалғанда уақыт қадамы елеулі роль атқармаса да, α – бөлшектің ұшуының траекториясы құрудың ұсынылған алгоритмінде α- бөлшектің атоммен соқтығысуын моментін уақытты түрде тіркеу үшін ол жеткілікті кішкене болуы керек.

Қадам онша кішкентай болмағаны жөн, бұл жағдайда траекторияны тұрғызуға біршама көп уақыт қажет болады.

Масштаб ретінде біз жүйеге тән параметрлерді алғандықтан, жүйенің өмір сүруінің типтік интервалы 1-ге тең уақыттың кішкене интервалы dt бірден әлдеқайда аз болуы керек, мысал үшін оны 0,001-ге тең деп аламыз. Таңдап алатын екінші параметр ретінде нысаналық параметрді алуға болады. Есептеу есептің меншікті масштабында жүргізілетіндіктен, нысаналық параметр бірден кішкене болуы керек екені түсінікті. Бірінші тәжірибе үшін p=0,5 деп алайық. "Счет-1" түймесін басып бір траекторияны тұрғызамыз.

Траектория күтілген талаптарға сәйкес болады. 1-кестеде траекторияның интегралдық сипаттамалары берілген: 1-бағанда есептелген траектория үшін нысаналық арақашықтық, екіншісінде θ шашырау бұрышы, үшіншіде бөлшектің атомға жақын келуінің ең кіші арақашықтығы, төртіншіде шашыраудың дифференциалдық S қимасы, бесінші бағанда нысаналық параметр өзгергенде шашырау бұрышының өзгеру шапшаңдығы қалай өзгеретін сипаттайтын шама келтіріледі. 2-кестеде α-бөлшектің кинематикалық энергияның ұшу уақытының әртүрлі моменттерінің мәні толтырылған. Уақыт моменттері кестенің жоғары жолында көрсетілген.

Шашырау бұрышы нысаналық параметр өзгергенде қалай өзгеретіні қарастырайық. Ол үшін нысаналық параметрлердің – 0,99-дан 0,99 аралығында үшін 11 траектория тұратын сериясын тұрғызамыз. P=1 мәндеріндегі траектория α бөлшек шашырайтын атомды шеткі нүктелерде жанасып, траекториясын өзгертпейтін жағдайға сәйкес келеді. Бұл траекториялар үшін шашыраудың дифферениалдық қимасы мен шашырау бұрышының өзгеру жылдамдығы анықталмаған, сондықтан оларды біз есептеп шығарып тастаймыз. 2.24-суретте тәжірибе өткізгеннен кейінгі экранның көрінісі берілген.



p	teta	Rmin	sigma	Dteta
0.500	120.000	1.000	0.250	132.363
-0.990	-16.219	1.000	0.256	793.065
-0.792	-75.254	1.000	0.250	187.496
-0.594	-107.117	1.000	0.250	142.379
-0.396	-133.343	1.000	0.250	124.764
-0.198	-157.160	1.000	0.250	116.894
-0.000	-180.000	1.000	0.000	359885.4(
0.198	157.160	1.000	0.250	116.918
0.396	133.343	1.000	0.250	124.823
0.594	107.117	1.000	0.250	142.510

p \ t=	0.000	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.000
0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
-0.990	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500

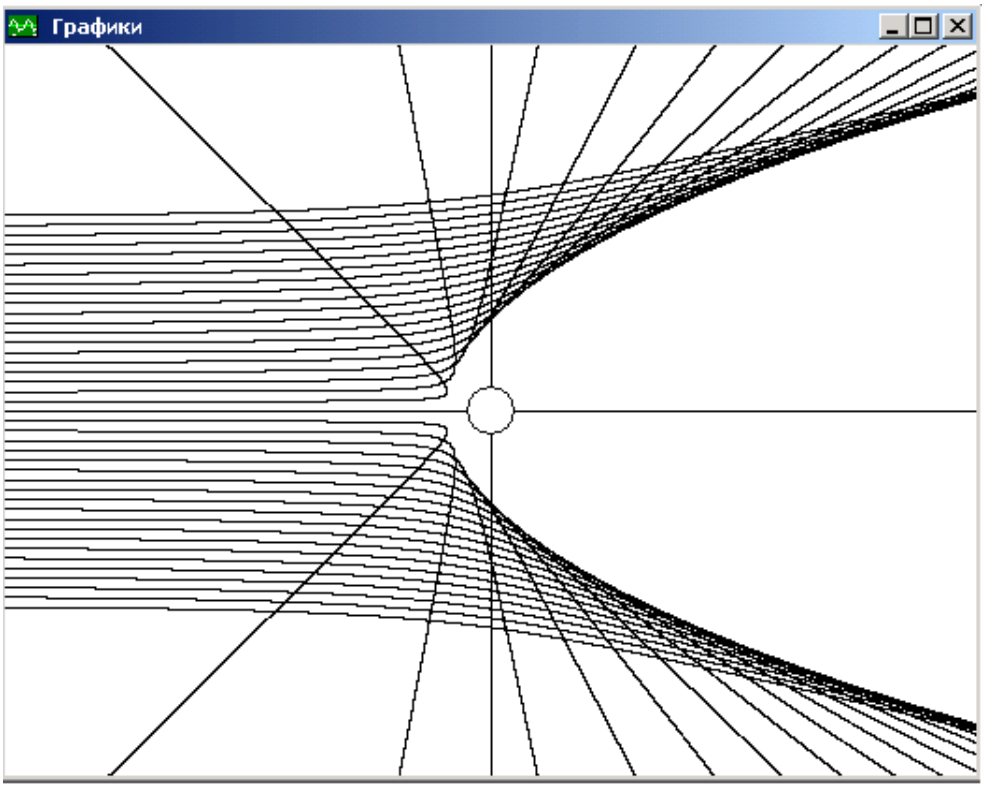
2.24- сурет. Томсон моделінде шашырау нәтижелері

Нысаналық параметрдің бірден кіші мәндерінде бөлшек барлық бұрыштарға бірқалыпты шашырайды. Мұны модулі бойынша бірден кіші нысаналық параметрлер үшін шашыраудың дифференциалдық қимасы бірдей болуы дәлелдейді. Күтілгендей бөлшектің кинетикалық энергиясы өзгермейді: бөлшектер күштің әсерінсіз ұшады, атомда серпімді соқтығысқа түседі.

α-бөлшектердің шашыраудың Кулондық шашырау α-бөлшектердің шашырауын тәжірибеде зерттейік. Томсон-Резерфорд жобасының негізгі терезесіндегі модельдер терезесінен Резерфорд тәжірибесін таңдап алайық. Бөлшектің жаңа орнын және жылдамдығын есептеу уақыт интервалын таңдап аламыз.Ол жеткілікті кішкене шама болғандықтан dt=0,01 деп алайық. Есептеу өлшемсіз шамаларда жүргіздіктен ол 1-ден кіші болуы керек, осы траекторияны тұрғызайық. Координаталар басында орналасқан радиусы 1 атом ядросын бейнелейтіндігін еске салайық. Траекторияның қаншалықты дәл есептелгендігін анықтау үшін dt=0,001 деп алып интегралдау қадамын азайтайық. Қисықтың түсін өзгертіп жаңа траектория тұрғызайық. Алынған екі қисықтың графиктері іс жүзінде сәйкес келеді, тек шашырау бұрышы үшінші мәнді цифрда өзгеше болады.

dt=0,001 деп алып, тағы тәжірибе жүргізейік. Енді шашырау бұрышының айырмасы төртінші таңбада болады. Тәжірибенің мұндай нәтижесін қанағаттанарлық деп есептеуге болады, қисықты тұрғызу уақыты да жеткілікті көп болмайды.

Шашырау заңын жүйелі түрде зерттеуге кірісеміз. Шашыраудың жалпы сипатын анықтау үшін нысаналық параметрдің 2-ден 2 дейінгі мәндері үшін 1 траекториядан тұратын серия тұрғызайық. Кестелермен экранды тазалаймыз. "Счетчик-М" түймесі арқылы траектория сериясын есептеу бағдарламасын іске қосамыз. Экранда оң және теріс бұрыштарға симметриялы шашырауды көреміз. Бөлшектердің бұрыштар бойынша шашырауы біркелкі емес. Шашыраудың дифференциалдық қимасының шашырау бұрышын тәелділігін дәлірек сипаттау үшін 2-ден 4-ке дейін және 4-тен 2- ге дейін нысаналық параметрлер үшін 11 траектория тұрғызамыз. Содан кейін бірнеше жекеленген траекториялар тұрғызамыз. 2.25-суретте бөлшектердің сериясын траекториялары келтірілген

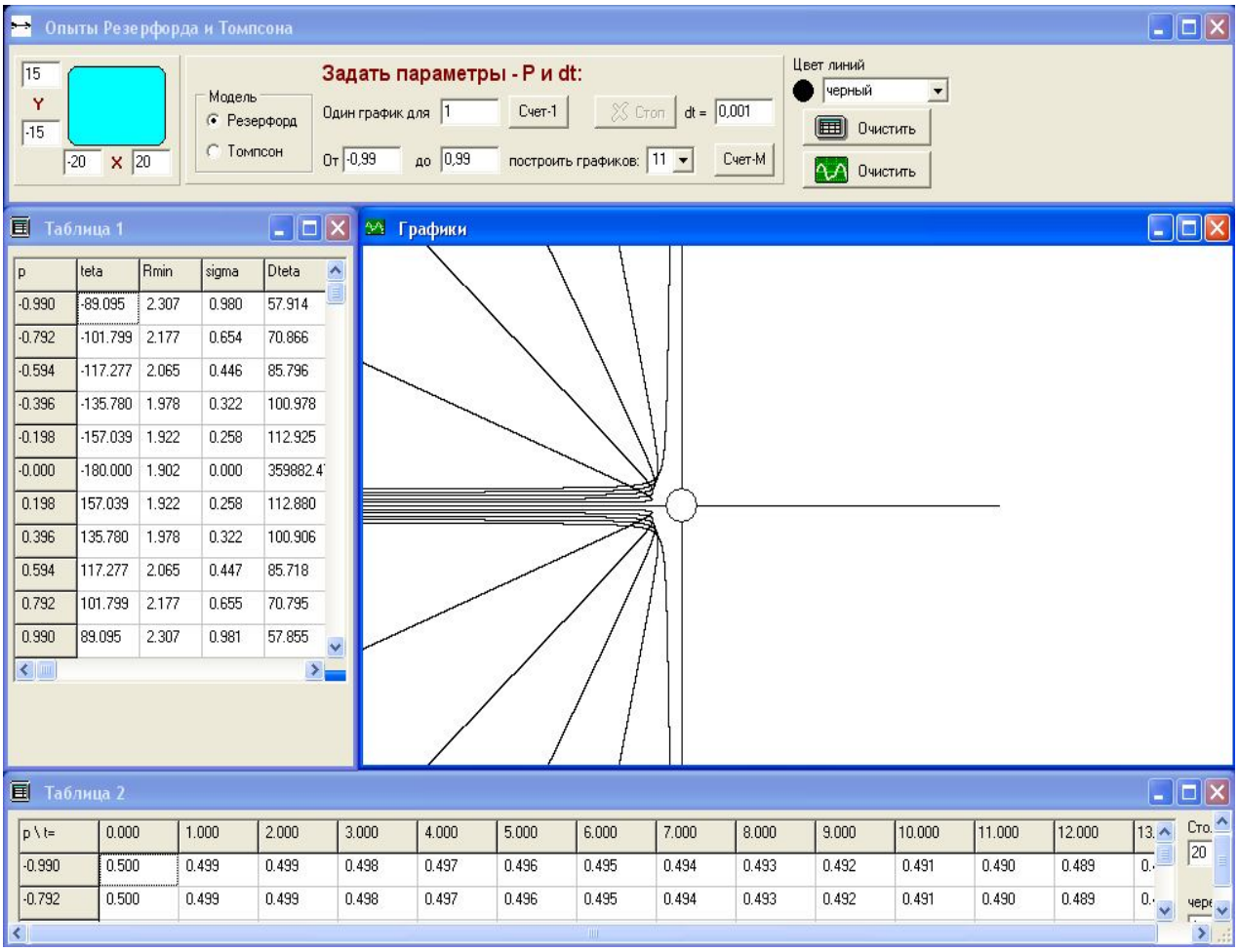


2.25-сурет. Кулон орталығындағы шашыратылған α-бөлшектерінің траекториясы.

Тәжірибе бөлшектердің көп бөлігі аз бұрыштарға шашырайтындығын көрсетеді. Үлкен бұрышқа нысаналық параметрі 1-ден кіші бөлшектер шашырайды. Бөлшектің ядроға жақындауының ең кіші мәні шамамен 2 бірлікке тең (ол салыстырмалы бірліктермен өлшенген).

Ядроға жақындаған сайын бөлшек тежеліп, оның кинетикалық энергиясы кемиді. Шашыраганнан кейін бөлшек электр өрісінде бастапқы жылдамдығына дейін үдейді.

"Итоговый график" терезесінде шашыраудың дифференциалдық қимасының бұрышына тәуелділігінің графигін тұрғызамыз. Графикте кішкентай бұрышқа шашырайтын бөлшектің санының елеулі өсетіні көрінеді. Біздің тәжірибелерде шашыраудың дифференциалдың қимасының өзгеру заңы Э. Резерфордың нақты тәжірибелерінің нәтижесімен сәйкес келеді. α-бөлшек ядроның Кулондық өрісінде шашырайды, ядроның өлшемі алынған өлшемдер бірлігінде 1-ге тең. 2.26-суретте Резерфордтың атом моделінде бөлшектердің шашырауы тәжірибесінің нәтижесі көрсетілген. Нысаналық параметрі 1-ден кем бөлшектер 90^0 немесе одан үлкен бұрышқа шашырайды. Бұл тәжірибе атомның планетарлық моделіне сәйкес келеді.



p	teta	Rmin	sigma	Dteta
-0.990	-89.095	2.307	0.980	57.914
-0.792	-101.799	2.177	0.654	70.866
-0.594	-117.277	2.065	0.446	85.796
-0.396	-135.780	1.978	0.322	100.978
-0.198	-157.039	1.922	0.258	112.925
-0.000	-180.000	1.902	0.000	359882.4
0.198	157.039	1.922	0.258	112.880
0.396	135.780	1.978	0.322	100.906
0.594	117.277	2.065	0.447	85.718
0.792	101.799	2.177	0.655	70.795
0.990	89.095	2.307	0.981	57.855

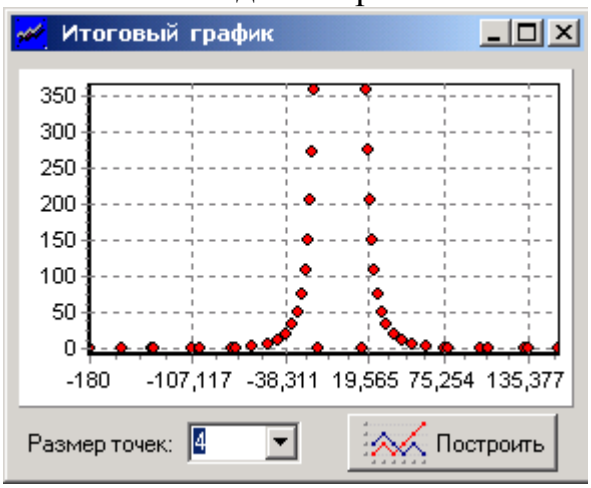
p \ t=	0.000	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.000	10.000	11.000	12.000
-0.990	0.500	0.499	0.499	0.498	0.497	0.496	0.495	0.494	0.493	0.492	0.491	0.490	0.489
-0.792	0.500	0.499	0.499	0.498	0.497	0.496	0.495	0.494	0.493	0.492	0.491	0.490	0.489

2.26 -сурет. Резерфорд моделіндегі шашырау нәтижелері.

Шашыраудың модельдерін салыстыру үшін алынған графиктерге Томсон моделіндегі қисықтарды қосып салайық. Ол үшін графиктер және кестелер алаңын тазаламай, Томсон моделіне көшеміз. Нысаналық –

параметрлердің – 0,99-дан 0,99-ға дейінгі мәндері аралығында траекториялар сериясын тұрғызамыз.

"Итоговый график" алаңында қисықтарды тұрғызамыз. 2.27 суретте екі модель үшін шашыраудың дифференциалдық қимасының шашырау бұрышынан тәуелділігі келтірілген. Томсон моделі үшін ол графиктің төменгі жағындағы горизонталь сызық.



2.27 - сурет. Екі модель үшін дифференциалдық қиманың шашырау бұрышынан тәуелділігі.

3. Оқу процессінде пайдалануға болатын ұсыныстар.

Осы жасалған виртуалдық зертханалық практикумда жалпы физика курсының 4 бөлімі бойынша 23 жұмыс қарастырылған. Механика бөлімі бойынша: тұтқырлы үйкеліс күші болғандағы дененің қозғалысын оқып үйрену; тербелу әдісімен біртекті дискінің инерттілік моментін анықтау; өзара перпендикулярлы тербелістерді есептеу (Лиссажу фигуралары); пружиналы маятниктің резонансын оқып үйрену; маятниктің байланысты тербелістерін оқып үйрену; орталық тартылыс өрісіндегі қозғалыс; нақты ортадағы көкжиек бұрышына лақтырылған дененің қозғалысы; физикалық маятник (шар-стержень жүйесі). Молекулярлық физика бөлімі бойынша: ауырлық күші өрісіндегі брокндық бөлшектердің таратылуын оқып үйрену және Больцманның тұрақтысын анықтау. Электр және магнетизм бөлімі бойынша: электростатикалық өрісті оқып үйрену; элементарлық зарядты (Милликеннің тәжірибесі) өлшеу; Ом заңын оқып үйрену; лампалық диодты оқып үйрену; парамагнетиктердің магниттенуін оқып үйрену; тербеліс контуры және резонанс. Оптика бөлімі бойынша: Фраунгофердің дифракциясын оқып үйрену; призманың сынуынының көрсеткішінің тәуелділігін оқып үйрену; фотоәсерді оқып үйрену; атом модельдері.

Қорыта айтар болсақ оқу процесінде пайдалануда төмендгіні меңгерді:

-«Физика», «Теориялық механика», «Қолданбалы механика», «Инженерлік механика», «Компьютерлік технологиялар», «Электр энергетикасындағы математикалық есептер мен компьютерлік модельдеу», «Математикалық әдістер мен өнеркәсіптегі модельдер» курстары бойынша зертханалдық жұмыстарды орындағанда;

-модельдеу бағдарламаларымен жұмыс жасау курстық немесе дипломдық жобаның неігізін қалай алады, себебі ол өз сипаты бойынша студент басты роль атқаратын кішігірім зерттеумен бірдей болып саналады;

4. ӘДЕБИЕТТЕР ТІЗІМІ.

1. Веденов А. А. Моделирование элементов мышления [Текст] / А.А Веденов ; М.: Наука, 2008. - 112 с.
2. Амосов Н. М. Моделирование мышления и психики. / Н.М Амосов М.: Наука, 2005. – 168 с.
3. Бальцук Н. Б. Некоторые возможности использования электронно-вычислительной техники в учебном процессе./ М.М.Буняев, В.Л. Матросов - М.: Прометей, 2010. - 138 с.
4. Кочергин А. Н. Моделирование мышления [Текст]./ /А.Н. Кочергин - М.: Наука, 2009. – 146 с.
5. Абдулла Х. Х. Численно-аналитические методы математического моделирования нелинейных обобщенно-механических систем в среде компьютерной математики Maple / Х. Х. Абдулла Автореф. дис. … канд. ф.-м. наук: 27.05.2011. – Казань: Татарский государственный гуманитарно-педагогический университет, 2011. – 20 с.
6. Федоренко Р. П. Введение в вычислительную физику: / Р. П. Федоренко Учеб. пособие: Для вузов. - М.: Моск. физ.-техн. ин-т, 2004. — 528 с.
7. Майер В. В Эксперимент на дисплее: Первые шаги вычислительной физики / В. В. Майер - М.: Наука, 2009.— 175 с.
8. Карлащук В. И. Электронная лаборатория на IBM PC. Программа Electronics Workbench и ее применение. / В. И. Карлащук Издание 2-е, дополненное и переработанное. –М.: Солон-Р, 2001. - 736с.
9. Останина А.М. Применение математических методов и ВМ [Текст] / А.М. Останина - Мн.: ДизайнПРО, 2005. – 496 с.
10. Куприенко В. Д. Педагогические программные средства: Методические рекомендации для разработчиков ППС / И. В. Мещерин , В. Д. Куприенко– Омск: ОГПИ им. А.М. Горького, 1991. – 182 с.
11. Тарасик В.П. Математическое моделирование технических систем. .[Текст]/ В.П. Тарасик – Мн.: Дизайн ПРО, 2007. – 640 с.
12. Фролов И.Т. Гносеологические проблемы моделирования. / И.Т. Фролов - М.: Наука, 2004. - 286 с.
13. Штофф В. А. Моделирование и философия. / В. А. Штофф - М.: Наука, 2006. – 164 с.
14. Щербаков Н.Р. Математическое и компьютерное моделирование динамического состояния систем передачи движения. Автореф. дис. … д-ра ф.-м. наук: 12.11.2009. – Томск: ГОУ ВПО «Томский государственный университет», 2009. – 30 с.
15. Кардашев Г.А. Виртуальная электроника. Компьютерное моделирование аналоговых устройств/ Г.А. Кардашев – М.: Горячая линия-Телеком, 2002. - 260 с.

16. Харченко Г.И. Компьютерные программы учебного назначения как средство активизации учебной деятельности студентов вуза / Г.И. Харченко Дис. … канд. пед. наук. – Ставрополь: Ставропольский государственный университет, 2005. – 205 с.
17. Галагер Р. Метод конечных элементов: Основы. / Р. Галагер - М.: Мир, 2004.- 428 с.
18. Гулд Х., Тобочник Я. Компьютерное моделирование в физике / Х. Гулд, Я Тобочник В 2-х частях. Часть 2. — М.: Мир, 2000. — 400 с.
19. Дульнев Г.Н., Парфенов В.Г., Сигалов А.В. Применение ЭВМ для решения задач теплообмена/ Г. Н . Дульнев , В. Г. Парфенов Учеб. пособие для теплофизич. и теплоэнергетич. спец. вузов. — М.: Высш. шк., 2000. — 207 с.
20. Иванов В. В. Методы вычислений на ЭВМ/ В. В. Иванов , Справочное пособие. — Киев: Наук думка, 2006—584 с.
21. Ильина В. А., Силаев П. К. Численные методы для физиков-теоретиков/ В. А. Ильина, П. К. Силаев — Москва-Ижевск: Институт компьютерных исследований, 2004. — 118с.
22. Бахвалов Л. Компьютерное моделирование: долгий путь к сияющим вершинам. / Бахвалов Л. Компьютерра. - 1997. - № 40. - С.15-25.
23. Ильина В. А. Компьютеры и нелинейные явления: Информатика и современное естествознание / В. А. Ильина - М.: Наука, 2008. — 192 с.
24. Берковский Б. М. Компьютеры, модели, вычислительный эксперимент. Введение в информатику с позиций математического моделирования / Б. М. Берковский — М.: Наука, 2008. - 176 с.
25. Коханер Д., Моулер К. Численные методы и программное обеспечение / Д. Коханер, К. Моулер — М.: Мир, 2001. —575 с.
26. Кунин С. Вычислительная физика / С. Кунин — М.: Мир, 1992. — 518 с.
27. Лаврентьев М.А., Шабат Б.В. Проблемы гидродинамики и их математические модели. — М.: Наука, 2003. —416 с.
28. Майер В. В. Свет в оптически неоднородной среде: учебные исследования / В. В. Майер — М.: Физматлит, 2007. — 232 с.
29. Мудров А. Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль / А. Е. Мудров — Томск: МП "РАСКО", 2001. — 272 с.
30. Поттер Д. Вычислительные методы в физике / Д. Поттер — М.: Мир, 1975. — 392 с.
31. Полянин А.Д. Справочник по нелинейным уравнениям математической физики: точные решения / А. Д. Полянин , В. Ф. Зайцев — М.: Физматлит, 2002- 432 с.
32. Ракитин В.И., Практическое руководство по методам вычислений с приложением программ для персональных компьютеров / В.Е. Первушин , В.И. Ракитин Учеб. пособие. — М.: Высшая шк., 2008. — 383 с.
33. Роуч П. Вычислительная гидродинамика. / П. Роуч М. : Мир, 1980.

— 616 с.
34. Рябенький В. С. Введение в вычислительную математику / В. С. Рябенький: Учеб. пособие. — М.: Физматлит, 2000. — 296 с.
35. Самарский А. А., Вабищевич П.Н., Самарская Е.А. Задачи и упражнения по численным методам: [Текст] / А. А . Самарский Учеб. пособие. — М.: Эдиторал УРСС, 2000. — 208 с.
36. Самарский А. А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры / А. А. Самарский — М.: Физматлит, 2001. — 320 с.
37. Тихонов А. Н., Самарский А. А. Уравнения математической физики / А. Н. Тихонов, А. А. Самарский — М.: Наука, 2006. —724 с.
38. Килин А. А. Разработка комплекса программ для компьютерного исследования динамических систем / А. А. Килин Автореф. дис. … д-ра ф.-м. наук: 01.07.2009. Ижевск: УдГУ, 2009. – 46 с.
39. Бубенчиков А. М., Щербаков Н.Р. Математическое моделирование динамики нового вида зацепления в передаточных механизмах / А. М. Бубенчиков Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. – С. 241–243.
40. Пономарёв С. М., Ховричева М.Л. Особенности лабораторного эксперимента в преподавании естественнонаучных дисциплин. Материалы конференции- выставки "Информационные технологии в образовании" / М. Л. Ховричева , С. М. Пономарёв - М.: МИФИ, 2000. - С. 137–141.
41. Бирюков Б.В., Моделирование / Ю. А. Гастеев , Е. С. Геллер - М.: БСЭ, 1994. - 252с.: ил
42. Гультяев А. Визуальное моделирование в среде MATLAB / А. Гультяев Учебный курс. - СПб.: Питер, 2000. - 432с.: ил.
43. Дименштейн Р.П., Яковлев А.Г. Информатика или компьютерное дело. // Информатика и образование. - 1989. - № 3. - С. 5-27.
44. Иванова А. Ю. Практическое моделирование. Компьютерный эксперимент. Лабораторный практикум / А. Ю. Иванова Учебное пособие. – Томск: Изд-во Том. гос. ун-т систем управления и радиоэлектроники, 2005. -236с.
45. Иванова А. Ю. Практическое моделирование. Компьютерный эксперимент. Методические указания для преподавателя / А. Ю. Иванова Томск: Изд-во Том. гос. ун-т систем управления и радиоэлектроники, 2005. -184с.
46. Церенова О. А. Математическое моделирование: Пособие для учителя / О. А. Церенова - Пермь: Перм. гос. пед. ун-т, 2005. - 259с.
47. Могилев А.В., Злотникова И.А. Элементы математического моделирования / И. А. Злотникова, А. В. Могилев – Омск: Изд-во ОМГПУ, 2005. - 104с.
48. Шестаков А.П. Математическое моделирование в вузе / А.П. Шестаков Информатизация образования – 93. Тез. докл. научно-практ.

конф. - Екатеринбург: Изд-во Уральского ГПУ, 2003. - С. 12-13.
49. Шестаков А. П. Преподавание курса "Математическое моделирование" в техническом вузе. Математическое моделирование систем и явлений / А. П. Шестаков Тез. докл. межрегиональной научно-техн. Конф – Пермь: Изд-во ПГТУ, 2003. - С. 8-13.
50. Трофимова Т. И. Курс физики: Учеб. пособие для вузов / Т. И. Трофимова–М.: Высш. шк., 2000. -478с.: ил.
51. Самсонов В. Е. Математическое моделирование движения тонкого слоя жидкости под действием поверхностных сил / В. Е. Самсонов Дис. … канд. ф.-м. наук. – Ставрополь: Ставропольский государственный университет, 2003–144 с.
52. Лапин В. Г. Математическое моделирование фронтальной части течения в каналах и реках при нестационарном стоке / В. Г. Лапин Дис. … канд. ф.-м. наук. – Ставрополь: Ставропольский государственный университет, 2005.–137 с.
53. Банько М. . Применение метода моментных уравнений для построения и исследования устойчивости математических моделей со случайными параметрами/ М. А. Банько Дис. … канд. ф.-м. наук. – Ставрополь: Ставропольский государственный университет, 2005. – 159 с.
54. Берковский Б. М., Полевиков В.К. Вычислительный эксперимент в конвекции / Б. М. Берковский - Мн.: Университетское , 2008. - 167 с.

5. ҚОСЫМШАЛАР

«**Тартудың центрлік өрісіндегі қозғалыс** » бағдарламасының фрагменті

```
{Параметрлер блогы}
const Nparam = 5; {таблицадағы парамтерлер саны}
    tabTitles: array [0..4] of string[2] = ('V0','e','T','a','k3');
    tab2Tit='V0 \ t=';
    NZ1=6; NZ2=3; {кестедегі он сандық белгілер}
    neopred=-1000;{белгінің мәні таблицада берілмеген}
    myScreenX=520; myScreenY=(3*myScreenX) div 4; {графикалық
беттердің белгісі}
    Narg=0; Nfunc=2; {бағандардың нөмірленуі}

{жеделдету}
function ax(x,y,t:real):double;
begin ax:=-x/(sqrt(x*x+y*y)*(x*x+y*y));end;
function ay(x,y,t:real):double;
begin ay:=-y/(sqrt(x*x+y*y)*(x*x+y*y));end;
{бастапқы шарттар}
procedure nachsost(Vnach:double);
begin t:=0;x:=1;y:=0;Vx:=0;V0:=Vnach;Vy:=V0 end;
{есептеудің бір қадамы және визуализациялауға шақыру}
procedure trace;
begin vx:=vx+ax(x,y,t)*dt;
    x:=x+vx*dt;
    vy:=vy+ay(x,y,t)*dt;
    y:=y+vy*dt;
    grPutPixel(x,y);
end;
{траекторияның интегралдық параметрлері}
procedure integr;
begin {param[0]:=v0;} param[1]:=e; param[2]:=tp; param[3]:=a; param[4]:=k3;
    tabLine; {кестенің жолдарын біріктіру}
end;
procedure graphic1(mnogo:boolean);
var v1,v2,v3,vh,s,Xmin,Ymax,yold,xold,Tprint,tpr:double;
   i,n:integer;
begin if mnogo then {көп сызбалар}
       begin v2:=StrToFloat(Form1.Edit2.Text); {басы}
           v3:=StrToFloat(Form1.Edit3.Text); {соңы}
           n:=Form1.ComboBox1.ItemIndex+2;{ траект. саны}
           vh:=(v3-v2)/(n-1);{шаг} v0:=v2;{баст. жылдамд}
```

```
  end   else {1 график}
  begin v1:=StrToFloat(Form1.Edit1.Text);
        v0:=v1;{ баст. жылдамд }
        n:=1;{1 траект.} vh:=0{қадам}
  end;
Xmin:=StrToFloat(Form1.Edit4.Text);{сол жақ шеті}
Ymax:=StrToFloat(Form1.Edit7.Text);{жоғарғы жақ шеті}
dt:=StrToFloat(Form1.Edit8.Text);  {уақыт қадамы}
Tprint:=StrToFloat(form4.edit2.Text);
myStop:=false; buttons(true); { "Стоп" белгісін белсендіру}
i:=1;  { траекторияның нөмірі}
repeat {n траекториялар}
      if v0<>0 then {V0=0 - есептемейді}
        begin ClearLastPoint; {соңғы нүктені тастау}
            nachsost(v0);{бастапқы шарттар} e:=0;
            {T, a и K3 әзірге анықталмаған}
            tp:=neopred; a:=neopred; k3:=neopred;
            tpr:=Tprint; s:=0; param[0]:=V0;
            repeat {траекторияны есептеу}
                xold:=x;yold:=y;{есте сақтау}
                t:=t+dt; trace; {қадам мен  визуализация}
                s:=s+(x*vy-y*vx); {сектор алаңы}
                if (t>=tpr) then
                    begin s:=0.5*s*dt; tpr:=tpr+Tprint;
                       tab2AddValue(s); s:=0
                  end;
                if (xold>0)and(x<=0) then e:=abs(y-1);
                if (yold>=0)and(y<0)then {анықтау T, a и K3}
                  begin tp:=2*t;a:=(1-x)/2;
                      k3:=tp*tp/(a*a*a);
                  end;
                {пауза для рисования}
                Application.ProcessMessages;
            until (((yold<0)and(y>0)){айналым аяқталды} or
                (((x<Xmin)or(y>Ymax))and(e>1)))
                {графиктің шетіне шығу }
                or myStop; {"Стоп" белгісін басу}
            integr; {траекторияның орталық параметрлері}
            tab2NextLine;
        end; {if v0<>0}
      v0:=v0+vh; i:=i+1; {келесі  траекторияға}
until (i>n) or myStop; {шығу}
```

```
    buttons(false); { "Счет" және "Стоп" белгісін қайта бастапқы күйге келтіру.}
end;
```

«**Нақты ортада көкжиекке бұрыш жасай лақтырылған дененің қозғалысы**» бағдарламасының фрагменті

```
{ private declarations }
  x,y,v,vx,vy,ax,ay,t,tup,tdown,angle,dt,M,k,ymax,xmax,vend,vup,ygol:
Extended;
  public
{ public declarations }
  end;

var
Form1: TForm1;
implementation
{ TForm1 }
procedure TForm1.btn_StartClick(Sender: TObject);
begin
 Self.DoubleBuffered:=True;
 v:= StrToFloatDef(le_Speed.Text,0);
 angle:= StrToFloatDef(le_Angle.Text,0);
 k:=StrToFloatDef(le_Trenie.Text,0);
 M:=StrToFloatDef(le_M.Text,1);
 x:=0;
 y:=0;
 vx:=v*cos(angle/180*pi);
 vy:=v*sin(angle/180*pi);
  t:=0;
 dt:=0.000001;
 Tmr.Enabled:=True;
end;

procedure TForm1.btn_StopClick(Sender: TObject);
begin
  Tmr.Enabled:=False;
end;
procedure TForm1.TmrTimer(Sender: TObject);
var i: Integer;
begin
for i:=0 to 100000 do
```

```
  begin
   ax:=-k/m*vx;
   ay:=-g-k/m*vy;
   vx:=vx+ax*dt;
   vy:=vy+ay*dt;
   x:=x+vx*dt;
   y:=y+vy*dt;
   t:=t+dt;
   while ymax<y do
    begin
    tup:=t;//көтерілу уақыты
    ymax:=y;//көтерілу биіктігі
    vup:=sqrt(sqr(vx)+sqr(vy));// жоғары нүктедегі жылдамдығы
    end;
   if y<0 then
    begin
    xmax:=x;/ /ұшу ұзақтылығы
    tdown:=t-tup;// түсу уақыты
    vend:=sqrt(sqr(vx)+sqr(vy));// соңғы жылдамдық
    ygol:=(180*arctan(vy/vx))/pi;// Vend және OX  oci арасындағы бұрыш
    Break;
    end;
  end;

Img.Canvas.Pixels[Round(x),Img.Height-Round(y)]:=clGreen;
 Img.Refresh;
 with memo1.lines do
  begin
   Clear;
   Add('X='+FloatToStr(x));
   Add('Y='+FloatToStr(y));
   Add('v='+FloatToStr(sqrt(sqr(vx)+sqr(vy))));
   Add('a='+FloatToStr(sqrt(sqr(ax)+sqr(ay))));
  end;
 if y< 0 then
  begin
   Tmr.Enabled:=False;
   Memo1.Lines.Add ('қозғалыс уақыты )
(теор)='+FloatToStr(2*v*sin(angle/180*pi)/g));
   Memo1.Lines.Add('қозғалыс уақыты (практ)='+FloatToStr(t));
   Memo1.Lines.Add('көтерілу уақыты ='+FloatToStr(tup));
   Memo1.Lines.Add('түсу уақыты ='+FloatToStr(tdown));
   Memo1.Lines.Add('Максималды көтерілу биіктігі ='+FloatToStr(ymax));
```

```
   Memo1.Lines.Add('ұшу ұзақтылығы ='+FloatToStr(xmax));
   Memo1.Lines.Add('жоғары нүктедегі жылдамдық ='+FloatToStr(vup));
   Memo1.Lines.Add('соңғы жылдамдық ='+FloatToStr(vend));
   Memo1.Lines.Add("соңғы жылдамдық арасындағы  бұрыш және OX
='+FloatToStr(ygol));
  end;
end;
initialization
  {$I unit1.lrs}
end.
```

«Физикалық маятник» бағдарламасының фрагменті

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, TeeProcs, TeEngine, Chart, Series, math;
const
  dt=0.0001;//квант уақыты
type
  TForm1 = class(TForm)
    img: TImage;
    graf: TChart;
    start: TButton;
    stop: TButton;
    exit: TButton;
    Timer: TTimer;
    chastota: TLabeledEdit;
    angle: TLabeledEdit;
    speed: TLabeledEdit;
    zatyxparam: TLabeledEdit;
    Series1: TLineSeries;
    Series2: TLineSeries;
    Series3: TLineSeries;
    vinsil: TLabeledEdit;
    chastvinsil: TLabeledEdit;
    procedure exitClick(Sender: TObject);
    procedure startClick(Sender: TObject);
    procedure TimerTimer(Sender: TObject);
    procedure stopClick(Sender: TObject);
  private
    { Private declarations }
    R,t,Phi,PhiDot,Phi2Dot,Omega,Delta,dphidot,f,chastF:extended;
```

```
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.startClick(Sender: TObject);
var i:integer;
begin
for i:=0 to 2 do //тазарту (бірнеше сынау үшін)
  graf.Series[i].Clear;
t:=0;
Phi:=StrToFloatDef(angle.Text,90)/180*pi;//бұрыш
phidot:=strtofloatdef(speed.Text,5);//жылдамдық
delta:=strtofloatdef(zatyxparam.Text,1)/5;//өшу параметрі
omega:=strtofloatdef(chastota.Text,5);//жиілік
f:=strtofloatdef(vinsil.Text,0);//күш
chastF:=StrToFloatdef(chastvinsil.Text,6);//еріксіз күштердің жиілігі
r:=min(Img.Width,img.Height) div 3;// радиус үшін қолданамыз 1/3
timer.Enabled:=true;// таймер іске қосамыз
end;
procedure TForm1.stopClick(Sender: TObject);
begin
timer.Enabled:=false;
end;
procedure TForm1.TimerTimer(Sender: TObject);
var xx,yy,i:integer;
begin
for i:=1 to 100 do //квант уақытының дәлдігіне байланысты
 begin
  dphidot:=-(2*delta*phidot+sqr(omega)*sin(phi)+f*sin(chastF*t)*cos(phi))*dt;
  phi2dot:=dphidot/dt;
  phi:=phi+(phidot+dphidot/2)*dt;
  phidot:=phidot+dphidot;
  t:=t+dt;// келесі уақыт моменті
 end;
{сызбаларды саламыз}
graf.Series[0].AddXY(t, phi);
graf.Series[1].AddXY(t, phi2dot);
graf.Series[2].AddXY(t, phidot);
with img do
 begin
```

```
  canvas.Pen.color:=clGreen;//обводка
  canvas.Brush.Color:=$D8E9EC;//фоновый цвет
  canvas.Rectangle(0,0,Width,Height);//прямоуголник на всё поле
  canvas.Pen.color:=clBlack;//цвет нити
 end;
xx:=Round(R*cos(3*pi/2+phi));//отступ по оси X
yy:=Round(R*sin(3*pi/2+phi));//отступ по оси Y
img.canvas.MoveTo((img.Height div 2),(img.Width div 2));//возвращаем значения в центр
img.canvas.LineTo((img.Height div 2)+xx,(img.Width div 2)-yy);//рисуем нить
img.canvas.Brush.Color:=clGreen;//цвет шарика
img.canvas.Ellipse(((img.Height div 2)+xx)+3,((img.Height div 2)-yy)+3,((img.Height div 2)+xx)-3,((img.Height div 2)-yy)-3);//рисуем шарик
end;
```

«Модели атома» бағдарламасының фрагменті

```
unit main;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons;
type
  TForm1 = class(TForm)
    grColor: TComboBox;
    Shape2: TShape;
    Label6: TLabel;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Label7: TLabel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
```

```
    ComboBox1: TComboBox;
    Label8: TLabel;
    Edit8: TEdit;
    Label9: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    RadioGroup1: TRadioGroup;
    procedure Button1Click(Sender: TObject);
    procedure grColorChange(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
{Блок параметров}
const Nparam = 5; { }
      tabTitles: array [0..4] of string[5] = ('p','teta','Rmin','sigma','Dteta');
      tab2Tit='p \ t=';
      NZ1=6; NZ2=3; { }
      neopred=-1000;{ }
      myScreenX=520; myScreenY=(3*myScreenX) div 4; { }
      Narg=1; Nfunc=3; { }
implementation
uses graph, tabl1, tabl2, rez_graph;
{$R *.DFM}
var x,y,Vy,Vx,t,tp,a,p,e,k3,dt:double; { }
    mnogo,myStop,Rezerford:boolean; { }
function ax(x,y,t:real):double;
begin if Rezerford
        then ax:=x/(sqrt(x*x+y*y)*(x*x+y*y))
        else ax:=0;
end;
function ay(x,y,t:real):double;
begin if Rezerford
        then ay:=y/(sqrt(x*x+y*y)*(x*x+y*y))
```

```
        else ay:=0;
end;
procedure nachsost(Pnach,Xnach:double);
begin t:=0;x:=Xnach;p:=pnach;y:=p;Vx:=1;Vy:=0;tab2AddValue(0.5)
end;
{есептеулдің бір қадамы мен визуализации}
procedure trace(gr:boolean);
begin vx:=vx+ax(x,y,t)*dt;
      x:=x+vx*dt;
     vy:=vy+ay(x,y,t)*dt;
      y:=y+vy*dt;
     if gr then grPutPixel(x,y);
end;
{ траекторияның интегралдық параметрлері}
procedure integr;
begin {param[0]:=v0;} param[1]:=e; param[2]:=tp; param[3]:=a; param[4]:=k3;
     tabLine; {сформировать строку в таблице}
end;
procedure buttons(b:boolean);
begin {активация кнопок "Счет" и "Стоп" ("в противофазе")}
with Form1 do
    begin BitBtn3.Enabled:=b;     {"Стоп"}
          Button1.Enabled:=not b; {"Счет" - 1 график}
          Button2.Enabled:=not b; {"Счет" - много графиков}
    end;
end;
procedure graphic1(mnogo:boolean);
var v1,v2,v3,vh,X0,s,Tprint,tpr:double;
   i,n:integer;
   gr:boolean;
begin Rezerford:=Form1.RadioGroup1.ItemIndex=0;
     X0:=-39;
     if mnogo then {көп }
       begin v2:=StrToFloat(Form1.Edit2.Text); {басы }
            v3:=StrToFloat(Form1.Edit3.Text); {соңы}
            n:=Form1.ComboBox1.ItemIndex+2;{число траект.}
            vh:=(v3-v2)/(n-1);{шаг} p:=v2;{нач. скорость}
       end   else {1 график}
       begin v1:=StrToFloat(Form1.Edit1.Text);
            p:=v1;{бастап. координата y}
            n:=1;{1 траект.} vh:=0{шаг}
       end;
     dt:=StrToFloat(Form1.Edit8.Text);  {шаг по времени}
```

```
Tprint:=StrToFloat(form4.edit2.Text);
myStop:=false; buttons(true); {активация кнопки "Стоп"}
i:=1;  {номер траектории}
repeat {n траекторий}
     ClearLastPoint; {сбросить коорд. последней точки}
          nachsost(p,x0);{начальные условия} e:=0;
          {T, a и K3 әлі белгісіз}
          tp:=-x0; a:=neopred; k3:=neopred;
          param[0]:=p;
           tpr:=Tprint;
           gr:=true;
          repeat { траекторияны есептеу}
               t:=t+dt; trace(gr); {қадам мен визуализация}
               a:=sqrt(x*x+y*y);   {орталыққа дейінгі қашықтық}
               if Rezerford
                  then begin if a<tp then tp:=a end
                  else begin if a<=1 then
                             begin x:=-sqrt(1-p*p);
                                  vx:=p*p-x*x;vy:=-2*p*x;
                                  tp:=1
                             end else
                               begin if a<tp then tp:=a; end;
                        end;
               if (t>=tpr) then
                  begin s:=0.5*(vx*vx+vy*vy); tpr:=tpr+Tprint;
                       tab2AddValue(s);
                end;
               {пауза }
               Application.ProcessMessages;
          until (a>abs(x0)+1){частица улетела далеко}
               or myStop; { кнопка "Стоп" басылып тұр}
         e:=arctan(vy/vx);if (p>0)and(e<0) then e:=e+pi;
          if (p<0)and (e>0)then e:=e-pi;
          {таратылуы нүктесін есептеу}
          p:=p+0.001; nachsost(p,x0); gr:=false;
          repeat {расчет траектории}
               {xold:=x;yold:=y;{есте сақтау}
               t:=t+dt; trace(gr); {қадм мен визуализация}
               a:=sqrt(x*x+y*y);{орталықтан қашықтығы}
          if not Rezerford
               then begin if a<=1 then
                          begin x:=-sqrt(1-p*p);
                               vx:=p*p-x*x;vy:=-2*p*x;
```

```
                    end;
                end;
              {пауза для рисования}
              Application.ProcessMessages;
            until (a>abs(x0)+1){бөлшек алысқа ұшып кетті}
                or myStop; { "Стоп" белгісі басылып тұр}
            s:=arctan(vy/vx);if (p>0)and(s<0) then s:=s+pi;
            if (p<0)and (s>0)then s:=s-pi;
            k3:=180*abs(e-s)/(pi*0.001);p:=p-0.001;
            if sin(e)<>0 then a:=0.001*p/abs(e-s)/sin(e)
                else begin a:=neopred; k3:=neopred end;
                e:=180*e/pi;
            integr; { траекторияның интегралдық параметрлері }
            tab2NextLine;
        p:=p+vh; i:=i+1; {к следующей траектории}
    until (i>n) or myStop; {выход когда все траект. или по кнопке}
    buttons(false); { "Счет" и "Стоп" бастаппқы күйге келтіру.}
end;
 procedure ShowWin;
begin form3.show; { 1 таблицасын көрсету}
    form2.show; {терезені көрсету}
    grSetSize(myScreenX,myScreenY); {өлшемін қою}
    centr_krug;
    form4.show; {показать окно с таблицей 2}
    form5.show;
    Form1.SetFocus; {фокусты бастапқы терезеге көшіру}
end;
procedure TForm1.Button1Click(Sender: TObject);
begin ShowWin; {показать окна}
    mnogo:=false; graphic1(mnogo); { 1 графикті салу}
end;
procedure TForm1.Button2Click(Sender: TObject);
begin ShowWin; {показать окна}
    mnogo:=true; graphic1(mnogo); {көп график салу}
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin  tabClearVal; {таблицаның мәнен кетіру 1}
     tab2ClearVal
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin {очистка графиков}
with form2.Image1 do Canvas.FillRect(Rect(0,0,width,height));
end;
```

```
procedure TForm1.BitBtn3Click(Sender: TObject);
{обеспечить досрочный выход из цикла расчетов}
begin myStop:=true
end;
procedure TForm1.grColorChange(Sender: TObject);
begin {выбор цвета графиков}
case grColor.ItemIndex of
    end;{case}
Shape1.Brush.Color:=grCurColor;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
with Form1 do {центровка главной формы}
    begin top:=0; left:=(screen.Width-width) div 2;
    end;
grColor.ItemIndex:=0; Shape1.Brush.Color:=clBlack; {выбор цвета}
combobox1.ItemIndex:=9; {число графиков}
end;
end.
```