**Коммерциялық емес акционерлік қоғам**

**АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ**

Электроника кафедрасы

# АСПАП ЖАСАУДАҒЫ КОМПЬЮТЕРЛІК ТЕХНОЛОГИЯЛАР

5В071600 – Приборлар жасау мамандығы студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Алматы 2014

ҚҰРАСТЫРУШЫ: С.Б. Абдрешова. Аспап жасаудағы компьютерлік технологиялар. 5В071600 – Приборлар жасау мамандығы студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар. – Алматы: АЭжБУ, 2014. – 33 б.

Әдістемелік нұсқаулықтар студенттерді, жүйелермен аспаптардың электрлік модельдерін, жылулықпен механикалық бірге қосқанда, орындауда қолданылатын MatLab математикалық модельдеу бағдарламасымен, таныстыруға арналған.

Әдістемелік нұсқауларда MatLab бағдарламасының жұмыс істеу принциптері, сонымен қатар алу азайту негіздері, жазықтықпен 3D-да функцияларды графикалық шағылыстыру, көрсету, матрицалық алгебра элементтері, оптимизациялау есептерімен теңдеулер жүйесін шешу, интегралдау әдістері және дифференциалдық теңдеулерді шешу көрсетілген.

Әдістемелік нұсқау дәрістік материалдарды пысықтау мақсатында құрастырылды және 5В071600 – Приборлар жасау мамандығының студенттеріне арналған.

Кесте – 3, сурет – 5, әдеб.көрсеткіші – 5 атау.

Пікір беруші: доцент Калиева С.А.

«Алматы энергетика және байланыс университеті» коммерциялық емес акционерлік қоғамының 2013 жылғы жоспары бойынша басылады.

© «Алматы энергетика және байланыс университеті» КЕАҚ, 2014 ж.

## Кіріспе

Бұл әдістемелік нұсқау математикалық аппаратты құрылғылар мен жүйелердің электрлік, жылулық және механикалық модельдерін құрастыру кезінде қажет, *Matlab* бағдарламалау ортасында машықтану үшін құрастырылған.

Әдістемелік нұсқаудың негізгі мақсаты – дәрістегі мәліметтерді пысықтау және *MatLab*-тың негізгі жұмыс істеу қағидаларын және есептеу негізін, функциялардың жазықтықтағы және 3D графикалық бейнеленуін, матрицалық алгебраның элементтерін, теңдеулер жүйесін және оңтайландыру есептерін шешу, интегралдау тәсілдерін және дифференциалдық теңдеулерді шешуді меңгеру.

Сондай-ақ, әдістемелік нұсқауда құрылғыларды және жүйелерді талдауға арналған *Matlab* бағдарламасының матрицалық логиканың қарапайым мысалдары келтірілген.

Әдістемелік нұсқау 5В071600 – Приборлар жасау мамандығына сай ҚР Мемлекеттік жалпыға міндетті білім беру стандартының талаптарына толықтай сәйкес келеді.

## 1 Зертханалық жұмыс №1. MATLAB жүйесімен тура есептеу режимінде жұмыс істеу

Жұмыс мақсаты: MATLAB жүйесінің қолданушы интерфейсін және тура есептеу режиміндегі жүйенің жұмыс негізін зерттеу.

### 1.1 Негізгі ережелер

MATLAB – ең көне және мұқият жетілдірілген математикалық есептеулерді автоматизациялайтын жүйелердің бірі. Ол әртүрлі есеп кластарын шығаруға оңай бейімдеуге болатын, кеңейтілген жүйе болып табылады. Оның бұлай (*MATrix LABoratory* – «матрицалық зертхана») аталу себебі – жүйе матрицалық және векторлық есептеулер жүргізуге бағдарланған.

*MATLAB* арнайы инженерлік есептеулер жүргізуге жасалған бағдарлама:

1) жүйенің математикалық аппараты қазіргі заманғы инженер мен ғалымдардың математикалық аппараттарына жақындатылған және векторлармен, матрицалармен, нақты және кешенді сандармен есептеуге сүйенген;

2) функционалдық тәуелділіктің графикалық бейнеленуі, инженерлік құжатта талап етілетін пішінде ұйымдастырылған.

MATLAB жүйесі көптеген пәрмендер мен операторлардың қатарын орындауға мүмкіндік береді. Пәрмендер деп – периферийлік құрылғыларды басқаратын құралдар, оператор деп – мәліметтермен (операндалармен) жұмыс

жасайтын құралдарды түсіну керек. Пәрмендер мен операторлар бағдарлама ішінен және тура есептеу режимінде де жасала алады.

Жүйемен тура есептеу режимінде жұмыс істеу диалогты түрде болады. Өрнекті есептеу пернетақтадан енгізу, ENTER батырмасын басу арқылы жүзеге асады. Бұл кезде қарапайым жолдық редактор жұмыс істейді.

Оның пәрмендері:

1) «→» – меңзерді бір белгіге оңға жылжыту;
2) «←» – меңзерді бір белгіге солға жылжыту;
3) «ctrl →» – меңзерді бір сөзге оңға жылжыту;
4) «ctrl ←» – меңзерді бір сөзге солға жылжыту;
5) «home» – меңзерді жолдың басына апару;
6) «end» – меңзерді жолдың соңына апару;
7) «↑ и ↓» – жолдарды үстіге немесе астыға жылжыту арқылы қарау;
8) «del» – меңзер тұрған белгіні өшіру;
9) «ins» – қою режимін өшіру/қосу.

Жүйенің маңызды пәрмендері:

1) HELP – көмек;
2) DEMO – көрсету;
3) INFO – ақпарат.

MATLAB бірнеше жүйелік айнымалыларды құрайды, оның ішінде:

1) pi – «пи» саны;
2) inf – машиналық шексіздік мәні;
3) ans – соңғы амалдың мәнін сақтайтын айнымалы.

Арифметикалық өрнектерде келесі операциялар белгісі қолданылады:

1) «+» – қосу;
2) «–» – азайту;
3) «*» – көбейту;
4) «/» – солдан оңға бөлу;
5) «\» – оңнан солға бөлу;
6) «^» – көрсеткішке шығару.

Әр элемент бойынша амалдар келесі түрде болады:

<амалдың белгісі>

MATLAB жүйесінде айнымалыны беру үшін меншіктеу операциясы қолданылады. Ол мына белгімен жазылады:

Айнымалы_аты = Өрнек.

Егер оператор жазуы «;» белгісімен аяқталмаса, онда қорытындысы пәрмендер терезесіне шығарылады, болмаса – шығарылмайды.

Егер операторда меншіктеу «=» белгісі болмаса, онда қорытынды мәні *ans* жүйелік айнымалыға меншіктеледі.

Сандармен жұмыс істегенде сандарды ұсыну пішінін беруге болады (бұл кезде барлық есептеулер шекті, яғни қос дәлдікпен жүргізіледі). Сандарды ұсыну пішінін орнату үшін «*format name*» пәрмені қолданылады, мұндағы «*name*» - пішіннің аты. «*Format*» пәрмені келесі шығару режимдерін орната алады:

1) format – сәйкесінше *format short*-қа ұқсас және әдеттегі орнату бойынша орнатылады;

2) format short – 5 белгісі бар белгіленген нүктелі пішін;

3) format short e – 5 белгісі бар жүзуші нүктелі пішін;

4) format long – 15 белгісі бар белгіленген нүктелі пішін;

5) format long e – 15 белгісі бар жүзуші нүктелі пішін;

6) format hex – он алты реттік пішін;

7) format + – шағын пішін, «+», «–» және «бос орын» оң, теріс және нөлдік элементтерді көрсету үшін қызмет етеді, жорамал бөлігін елемейді;

8) format bank – ақша бірлігіне арналған пішін;

9) format compact – келесі жолға өткізбейтін пішін;

10) format loose – пішін, *format compact*-қа кері.

*MATLAB* жүйесі нақты сандармен, сондай-ақ $z = Re(z) + i*Im(z)$ түрдегі комплекс сандармен жұмыс істейді, мұндағы $i$ (және $j$) – жорамал бөлігі; яғни -1 алынған квадрат түбір, $Re(z)$ – $z$ комплекс санының нақты бөлігі, ал $Im(z)$ – оның жорамал бөлігі. Комплекс сандармен жұмыс істеместен бұрын $i = sqrt(-1)$ немесе $j = sqrt(-1)$ анықтап алу керек. *real(z)* функциясы *Re(z)* комплекс санының нақты бөлігін қайтарады, ал *imag(z)* – жорамал *Im(z).* Комплекс санның модулін анықтау үшін *abs(z)* функциясы, ал аргументін анықтау үшін *angle(z)* функциясы пайдаланылады.

*MATLAB* жүйесі әртүрлі математикалық функцияларды есептеуге мүмкіндік береді. Келесі элементар алгебралық функциялар аргумент ретінде бір немесе екі нақты ($x$, $y$) немесе бір комплекс ($z$) сан қабылдайды:

1) abs(z) – z комплекс санының модулін есептеу немесе z нақты санының абсолют мәнін анықтау;

2) angle(z) – z аргументін есептеу;

3) sqrt(z) – квадрат түбірді есептеу;

4) real(z) – z -тің нақты бөлігін анықтау;

5) imag(z) – z -тің жорамал бөлігін анықтау;

6) round(z) – бүтін санға дейін дөңгелету;

7) fix(z) – нөлдің жағына қарай, ең жақын бүтінге дейін дөңгелету;

8) floor(z) – теріс шексіздік жаққа қарай, ең жақын бүтінге дейін дөңгелету;

9) sign(z) – белгінің функциясын есептеу;

10) rem(x, y) – x-ті y-ке бөлгендегі қалдықты анықтау;

11) exp(z) – z дәрежедегі e санын есептеу;

12) log(z) – z санының натурал логарифмін анықтау;

13) log10(z) – z санының ондық логарифмін анықтау.

MATLAB жүйесі келесі тригонометриялық және кері тригонометриялық функцияларды есептеуге мүмкіндік береді:

1) sin(z) – синусты есептеу;

2) cos(z) – косинусты есептеу;

3) tan(z) – тангенсті есептеу;

4) asin(z) – арксинусты есептеу;

5) acos(z) – арккосинусты есептеу;
6) atan(z) – арктангенсті есептеу;
7) atan2(y, x) – нүктенің координатасы бойынша арктангенсті есептеу.

## 1.2 Зертханалық жұмыстың жасалу реті

1.2.1 Айнымалы мәндерін есептеу және шығару.

Айнымалы мәндерін есептеу және 1.1 кестеге берілген пішіндегі мәндерін шығару.

1.1 кесте – Тапсырма нұсқалары

| № | Тапсырма | Пішін |
|---|---|---|
| 1 | abs(sin(0)+cos(0)+log(1))/exp(1) | format short |
| 2 | (log(1)+exp(10))/(sin(pi)–cos(pi)) | format short |
| 3 | (log(10)–exp(5))/(sin(2*pi)–cos(2*pi)) | format short |
| 4 | sign(sin(0)–cos(0)+log(1)+exp(1)) | format short |
| 5 | (sin(3*pi)+cos(3*pi)/(log(10)–exp(1)) | format short |
| 6 | abs(sin(0)+cos(0)+log(1))/exp(1) | format short e |
| 7 | (log(1)+exp(10))/(sin(pi)–cos(pi)) | format short e |
| 8 | (log(10)–exp(5))/(sin(2*pi)–cos(2*pi)) | format short e |
| 9 | sqrt(sin(0)–cos(0)+log(1)+exp(1)) | format short e |
| 10 | (sin(3*pi)+cos(3*pi)/(log(10)–exp(1)) | format short e |
| 11 | abs(sin(0)+cos(0)+log(1))/exp(1) | format long |
| 12 | (log(1)+exp(10))/(sin(pi)–cos(pi)) | format long |
| 13 | (log(10)–exp(5))/(sin(2*pi)–cos(2*pi)) | format long |
| 14 | sign(sin(0)–cos(0)+log(1)+exp(1)) | format long |
| 15 | (sin(3*pi)+cos(3*pi)/(log(10)–exp(1)) | format long |
| 16 | abs(sin(0)+cos(0)+log(1))/exp(1) | format long e |
| 17 | (log(1)+exp(10))/(sin(pi)–cos(pi)) | format long e |
| 18 | (log(10)–exp(5))/(sin(2*pi)–cos(2*pi)) | format long e |
| 19 | sqrt(sin(0)–cos(0)+log(1)+exp(1)) | format long e |
| 20 | (sin(3*pi)+cos(3*pi)/(log(10)–exp(1)) | format long e |

1.2.2 Комплексті айнымалылар.

*z* комплекс санын енгізу мысалы.

```
>> z= 1+2i
```

Көшіру, *Matlab*-қа қою және *Enter*-ді басу.

```
z=
  1.0000 + 2.0000i
```

i немесе j жорамал бөлігін белгілеп алуға болады.

```
>> z+j
ans=
  1.0000 + 3.0000i
```

```
>> i*z
   -2.0000 + 1.0000i
>> (2 - 4i)*(1 + z)
ans=
   12.0000 - 4.0000i
>> c= complex(3,-4)
c=
   3.0000 - 4.0000i
```

Комплекс-түйіскен санды есептеу.

```
>> conj(c)
cs=
   3.0000 + 4.0000i
```

С комплекс санының нақты және жорамал бөлігін бөліп алу.

```
>> a= real(c)
a=
    3
>> b= imag(c)
b=
    -4
```

Комплекс санның бұрышын және модулін анықтау.

```
>> md= sqrt(a^2 + b^2)
md=
   25
>> rd= atan2(-4,3)
rd=
   0.9273
>> gr= (180/pi)*rd
gr=
   -53.1301
```

### 1.3 Есеп беру мазмұны

1) Жұмыстың мақсаты.
2) Мәліметтерді есептеу және шығару.
3) Комплексті айнымалылары бар пәрмендерді сипаттау.
4) Қорытынды.

### 1.4 Бақылау сұрақтары

1) MATLAB-ты жүктеген кезде шығатын шақыру қандай түрде болады?

2) Тура есептеу режимінде жұмыс істеуге арналған MATLAB-тың негізгі пәрмендері қандай?

3) Сандардың пішіні қандай пәрменмен орнатылады?

4) Format hex пәрмені не үшін қызмет етеді?
5) Conj() пәрмені не үшін қызмет етеді?
6) Atan() және atan2() пәрмендерінің айырмашылығы қандай?
7) Real() пәрмені не үшін қызмет етеді?
8) Imag() пәрмені не үшін қызмет етеді?

## 2 Зертханалық жұмыс № 2. MATLAB жүйесінде векторлар мен матрицаларға амалдар қолдану

Жұмыстың мақсаты*:* MATLAB жүйесінде векторлар мен матрицалармен жүргізілетін негізгі амалдарды жүргізуді оқып үйрену.

### 2.1 Негізгі ережелер

MATLAB – векторлармен, матрицалармен және көпмүшелермен күрделі есептеулер жүргізуге арналған жүйе.

Векторлар мен матрицалардың мәндерін енгізгенде, квадрат жақшаға алып жазамыз. Бағандарды бөлу үшін бос орын пернесі, жолдарды бөлу үшін «;» белгісі қолданылады. MATLAB жүйесі мәні арифметикалық прогрессия болып табылатын векторды енгізуге мүмкіндік береді.

<Аты> = <БМ>:<Ш>:<СМ>,

мұндағы <БМ> – прогрессияның бастапқы мәні, <Ш> – прогрессия айырымы, <СМ> – соңғы мәні.

Сондай-ақ векторлар мен матрицалардың арифметикалық өрнек түріндегі, кез келген жүйедегі қол жетімді функциялардан құралатын элементтерді енгізуге мүмкіндік бар. Комплекс элементтері бар векторлар мен матрицаларды да беруге болады.

Кіші өлшемді матрицаларды үлкен өлшемді матрицалардың элементі ету арқылы матрицаларды кеңейтуге болады.

Келесі матрицалық функциялар кейбір кеңінен таралған матрицалар генерациясын қамтамасыз етеді:

1) zeros(m,n) – нөлдік элементтері бар матрица генерациясы;
2) ones(m,n) – бірлік элементтері бар матрица генерациясы;
3) rand(m,n) – кез келген мән қабылдайтын элементтері бар матрица генерациясы;
4) eye(m,n) –диагональда бірлік элементтері бар матрица генерациясы (m – жол саны; n – матрицадағы баған саны).

Мына өрнекті пайдаланып, матрицаны нөлдік өлшемге әкелуге болады:

Матрица_аты = [].

Мұнда матрицаның аты сақталып қалады, кейін оны кеңетіп қолдануға болады.

Матрицаны жою келесі пәрменмен орындалады:

clear Матрица_аты.

(а) матрицасының элементтерін бөліп көрсету келесі пәрмендермен орындалады:

1) a(i,j) – i жолындағы, j бағанында орналасқан элементті бөліп алу;
2) a(i,:) – i жолын бөліп алу;
3) a(:,j) – j бағанын бөліп алу.

MATLAB-та векторлар мен матрицаларға келесі амалдарды қолдануға болады:

1) + – қосу;
2) – – азайту;
3) * – көбейту;
4) / – бөлу амалы.

Егер бөлгіш те, бөлінді де скаляр шама болса, оның нәтижесі – скалярды скалярға бөлгендегі бөлінді. Егер бірінші операнда матрица, ал екіншісі скаляр болса, онда нәтижесінде әрбір элементі бөлінді скаляр болып табылатын матрица. Егер операндалар бірдей өлшемді квадрат матрицалар болса, онда нәтижесі – бірінші операнда матрицасын екінші операнда матрицасының керісіне көбейтінді болады.

\ - сол жақты бөлу амалы. Бірдей өлшемді квадрат матрицалар операнда болып табылады. Нәтижесі бірінші операндаға кері матрицаның екінші операнда матрицаға бөлгендегі матрица болып табылады.

' – транспонирлеу;

^ – көрсеткішке шығару;

inv(m) – m матрицасының керісін есептеу

pinv(m) – m матрицасының псевдоаударылғаны;

sqrtm(m) – матрицалық квадрат түбір;

poly(m) – матрицаның сипаттамалық көпмүшесінің коэффициенттері бар вектор;

det(m) – m матрицасының анықтауышының мәні;

trace(m) – m матрицасының ізі;

rank(m) – m матрицасының рангы.

MATLAB жүйесінде матрицалық немесе векторлық формада берілген мәліметтерді өңдеуге арналған көптеген функциялар бар.

*size(m)* функциясы *m* матрицасының жол және баған санын анықтауға арналған. Ол осы мәліметтерді құрайтын *[n, p]* векторын шығарады.

*max(v)* функциясы *v* векторының элементтер мәндерінің максимальды мәнін шығарады. Егер оның аргументі матрица болса, мысалы, *max(m),* онда әрбір бағандағы максималь элементтерін құрайтын, вектор-жолды шығарады.

*min(m)* функциясы ұқсас жұмыс жасайды, минимальды мәндерді құрайтын элементтерді бөліп алады.

*mean(v)* функциясы *v* векторының элементтерінің орташа мәнін қайтарады, ал *mean(m)* матрицалық аргументі бар әрбір баған мәліметтерінің орташа мәнінен тұратын жол-векторын қайтарады.

*std(v)* функциясы статикалық параметр қайтарады - v векторымен берілген, бірлік мәліметтер массиві үшін стандарт (квадрат) ауытқу.

Матрицалық аргумент үшін бұл функция әрбір баған үшін стандарт ауытқу жол-векторын қайтарады.

*sort(v)* сорттау функциясы элементтері мәндерінің өсу реті бойынша орналасқан вектор қайтарады. Матрицалық аргумент үшін бұл функция әрбір баған элементтері сортталған матрица қайтарады.

*sum(v)* функциясы *v* векторының элементтерінің қосындысын қайтарады, ал матрицалық аргумент үшін *sum(m)* функциясы әрбір баған элементтерінің қосындысын құрайтын жол-векторын қайтарады.

*prod(m)* функциясы алдыңғысына ұқсас әрбір баған элементтерінің көбейтіндісін құрайтын жол-векторын қайтарады.

## 2.2 Зертханалық жұмыстың жасалу реті

2.2.1 Пернетақтадан матрицалар мен векторларды енгізу.

Енгізу керек:

– (*v*) вектор-жолын;

```
>> v= [1 2 3]
v =
    1     2     3
```

– (*w*) вектор-бағанын;

```
>> w= [3;2;1]
w =
    3
    2
    1
```

– (*m*) матрицасын.

```
>> m= [1 2 3;3 2 1;1 3 2]
m =
    1     2     3
    3     2     1
    1     3     2
```

2.2.2 Арнайы түрдегі матрицалардың генерациясы.

Құру керек:

1) (*m0*) нөлдік элементтері бар матрицаны;

```
>> m0= zeros(3,3)
m0 =
    0     0     0
    0     0     0
    0     0     0
```

2) (*m1*) бірлік элементтері бар матрицаны;

```
>> m1= ones(3,3)
m1 =
    1     1     1
```

```
     1     1     1
     1     1     1
```

3) (*mr*) кез келген мәндер қабылдайтын элементтері бар матрица;

```
>> mr= rand(3,3)
mr =
   0.8147    0.9134    0.2785
   0.9058    0.6324    0.5469
   0.1270    0.0975    0.9575
```

4) (*me*) негізгі диагональда бірлік элементтері бар матрица;

```
>> me= eye(3,3)
me =
     1     0     0
     0     1     0
     0     0     1
```

5) матрицадан баған-векторын шығару;

```
>> m(1:3,2)
ans=
      2
      2
      3
```

6) матрицадан жол-векторын шығару;

```
>> m(3,1:3)
ans=
      1   3   2
```

7) матрица минорларын шығару;

```
>> m(1:2,1:2)
ans=
      1   2
      3   2
```

2.2.3 *mm* матрицасын есептеу**.**

2.1 кесте – Тапсырмалар нұсқасы

| № | Тапсырма |
|---|---|
| 1 | mm=v*w+m+mr*me |
| 2 | mm=m+mr*me |
| 3 | mm=v/m+mr*me |
| 4 | mm=w/m+mr*me |

Мысалы, №1 нұсқаның mn матрицасын анықтайық.

```
>> mm=v*w+m+mr*me
mm =
  11.8147   12.9134   13.2785
  13.9058   12.6324   11.5469
```

```
  11.1270   13.0975   12.9575
```

mm матрицасының жол және баған санын анықтау.

```
>> size(mm)
ans =
    3     3
```

mm матрицасының әрбір бағанының максималь элементін анықтау.

```
>> max(mm)
ans =
  13.9058   13.0975   13.2785
```

mm матрицасының әрбір бағанының минималь элементін анықтау.

```
>> min(mm)
ans =
  11.1270   12.6324   11.5469
```

mm матрицасының әрбір бағанындағы элементтерді қосу.

```
>> sum(mm)
ans =
  36.8475   38.6433   37.7829
```

mm матрицасының әрбір бағанындағы элементтерді көбейту

```
>> prod(mm)
ans =
 1.0e+003 *
   1.8281    2.1366    1.9867
```

sort(v) сорттау функциясы - v векторының элементтері мәндерінің өсу реті бойынша орналасады.

```
>> v=[123 11 66]
v =
  123    11    66
>> sort(v)
ans =
   11    66   123
```

2.2.4 Сызықты теңдеулер жүйесін шешу.

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 10 \\ -x_1 + x_2 + 2x_3 = 1 \\ x_1 - 2x_2 + 5x_3 = 4. \end{cases}$$

Теңдеулер жүйесін матрицалық түрде былай көрсетуге болады: $A\vec{x} = \vec{b}$ немесе $\vec{x} = A^{-1}\vec{b}.$

```
>> A=[1 2 3;-1 1 2;1 -2 5]
A=
   1   2   3
  -1   1   2
```

```
   1  -2   5
>> b=[10;1;4]
b=
   10
    1
    4
```

$A^{-1}$ кері матрицасын A детерминанты нөлден басқа (нөлге бөлуге болмайды) азғындалмаған матрицасынан ғана алуға болады.

```
>> det(A)
ans=26
```

Есептеудің бірінші түрі.

$A^{-1}$ кері матрицасы inv(A) пәрменімен анықталады.

```
>> A1= inv(A)
A1=
    0.3462   -0.6154    0.0385
    0.2692    0.0769   -0.1923
    0.0385    0.1538    0.1154
```

Кері матрицаның анықталуының дұрыстығын тексеру үшін, оның мына қасиетін пайдаланамыз:

$$A \cdot A^{-1} = E.$$

```
>> A*A1
ans=
   1.0000         0         0
        0    1.0000    0.0000
   0.0000         0    1.0000
>> x= A1*b
x=
   3
   2
   1
```

Есептеудің екінші жолы.

A1 матрицасының керісін есеп, b векторына (сол жағынан) көбейтудің орнына сол жақты бөлу амалын \ қолдануға болады.

```
>> x= A\b
x=
   3
   2
   1
```

Теңдеулер жүйесінің шешімі: $x_1=3$; $x_2=2$; $x_3=1$. Есептеуде екінші жол дұрысырақ екені анық.

### 2.3 Есеп беру мазмұны

1) Жұмыс мақсаты.

2) 2.2.1-2.2.3 пункттерінің нәтижесі.
3) Қорытынды.

### 2.4 Бақылау сұрақтары

1) Жол-векторын енгізу қалай жүргізіледі?
2) Баған-векторын енгізу қалай жүргізіледі?
3) Матрицаның жол және баған саны қалай анықталады?
4) Матрицаны max-ды және min-ды элементін анықтау үшін қандай амалдар қолданылады?
5) Матрица элементтерін қосу және көбейту амалдары қалай орындалады?
6) Matlab жүйесінде орындалатын басқада амалдарды атаңыз.
7) Матрицалар генерациясын қамтамасыз ететін функцияларды атаңыз.
8) Matlab жүйесінде матриалық немесе векторлық формада берілген мәліметтерді өңдеуге арналған функцияларды атаңыз.

## 3 Зертханалық жұмыс №3. MATLAB графикалық құралдары

Жұмыс мақсаты*:* MATLAB жүйесінің негізгі графикалық амалдарын оқып үйрену және гравикалық шығаруды іске асыратын бағдарламалар құру.

### 3.1 Негізгі ережелер

MATLAB-та есептеулер нәтижесін графикалық түрде көрсету үшін арнайы пәрмендер жиынтығы қолданылады.

Графиканың негізгі амалдары:

1) plot(x,y) – сызықтық масштабта график құру;
2) loglog – логарифмдік масштабта график құру;
3) semilogx – жартылогарифмдік (log x осі бойымен) масштабта график құру;
4) semilogy – жартылогарифмдік (log y осі бойымен) масштабта график құру;
5) polar – полярлы координаталар жүйесінде график құру;
6) mesh(x,y,z) – үшөлшемді жазықтықта график құру;
7) contour – контурлық сызықтармен – бірдей биіктік деңгейінде график тұрғызу;
8) bar(x,y) – бағандық гистрограмманың графигін тұрғызу;
9) stairstep – сатылық сызық түрінде график тұрғызу.

Графиктерді безендіру үшін келесі амалдарды пайдаланады:

1) text – графиктің берілген жеріне жазу шығару;
2) title('y=sin(x)') – титулдық жазу беру;
3) xlabel('x') – x осі бойынша жазу;
4) ylabel('y') – y осі бойынша жазу;

5) grid on – пунктирлі масштабты торды беру.

Графикалық амалдарды бағдарламалау кезінде келесі амалдар қолданылады:

1) axis([0 10 -3 3]) – X осі бойынша масштаб: 0-ден 10-ға, У осі бойынша: -3-тен 3-ке дейін;

2) hold on – адындағы графикті бір терезеде сақтау;

3) prtsc – графикалық экрандағы суретті баспадан шығару;

4) subplot(m,n,p) – терезені одан кішкетай терезелерге бөлу (m – вертикаль бойынша терезелер саны, n – горизонт бойынша, p – кіші терезе нөмірі).

### 3.2 Зертханалық жұмыстың орындалу реті

3.2.1 Функциялар графиктерін шығару.

3.1 кесте – тапсырмалар нұсқасы

| № | f1 | f2 | f3 | f4 |
|---|---|---|---|---|
| 1 | sin(x) | cos(x) | $x^2$ | ln(x) |
| 2 | $e^x$ | $x^2$ | x | \|x\| |
| 3 | sin(x)+cos(x) | cos(x)+ $x^2$ | $x^2$+ln(x) | ln(x)+ $e^x$ |
| 4 | sin(x)+$e^x$ | sin(x)+$x^2$ | sin(x)+x | sin(x)+\|x\| |
| 5 | x*sin(x) | x*cos(x) | $x^2$ | x*ln(x) |
| 6 | x*$e^x$ | sin(x)+$x^2$ | sin(x)+x | sin(x)+\|x\| |
| 7 | sin(x)*cos(x) | cos(x)*$x^2$ | $x^2$*ln(x) | ln(x)*$e^x$ |
| 8 | sin(x)*$e^x$ | sin(x)*$x^2$ | sin(x)*x | sin(x)*\|x\| |
| 9 | $\sin^2$(x) | $\cos^2$(x) | x | $\ln^2$(x) |
| 10 | sin(x)*$e^x$ | sin(x)*$x^2$ | x sin(x) | \|x\|*sin(x) |

Ескерту: y = abs(x) - |x| аргументінің модулін анықтайды.

subplot() пәрмені көмегімен бір графикалық терезенің 4 кіші терезесінде 3.1 кестесіндегі f1-f4 бағандарында орналасқан берілген функцияларды беунелеу.

1 терезе:

```
y1=f1(x)
y2=2*f1(x)
y3=3*f1(x)
```

2 терезе:

```
y1=f2(x)
y2=2*f2(x)
y3=3*f2(x)
```

3 терезе:

```
y1=f3(x)
y2=2*f3(x)
y3=3*f3(x)
```

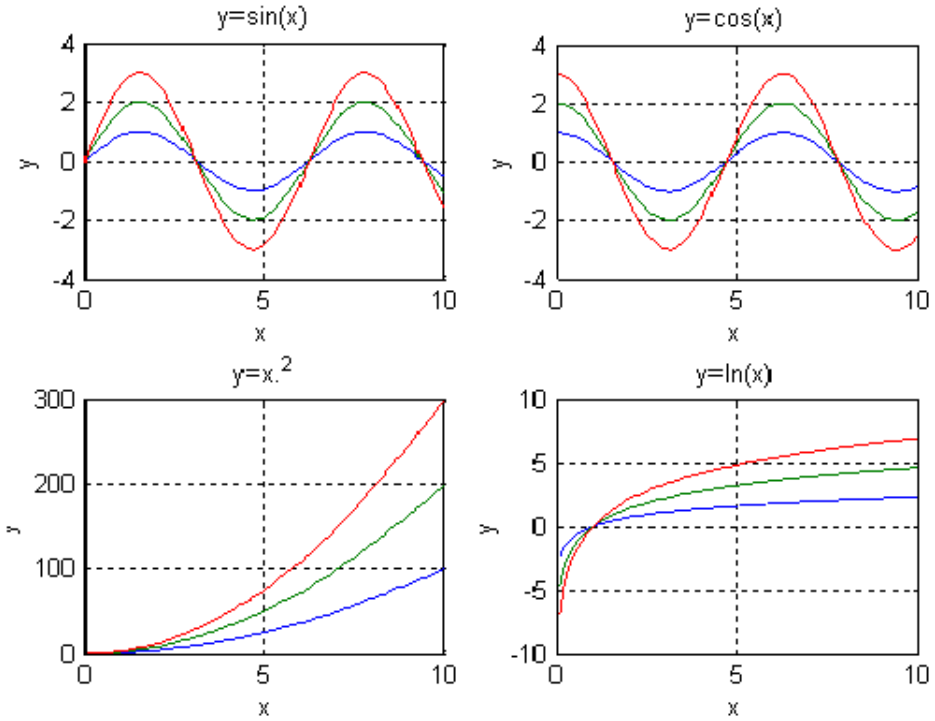
4 терезе:

```
y1=f4(x)
y2=2*f4(x)
y3=3*f4(x)
```

Үлгі:

```
>> x=0:0.1:10;
>> y1=sin(x);
>> y2=2*sin(x);
>> y3=3*sin(x);
>> axis([0 10 -3 3]);
>> subplot(2,2,1);
>> plot(x,y1,x,y2,x,y3);
>> title('y=sin(x)')
>> xlabel('x');
>> ylabel('y');
>> grid on
>> x=0:0.1:10;
>> y1=cos(x);
>> y2=2*cos(x);
>> y3=3*cos(x);
>> axis([0 10 -3 3]);
>> subplot(2,2,2);
>> plot(x,y1,x,y2,x,y3);
>> title('y=cos(x)')
>> xlabel('x');
>> ylabel('y');
>> grid on
>> x=0.1:0.1:10;
>> y1=x.^2;
>> y2=2*x.^2;
>> y3=3*x.^2;
>> axis([0 10 0 300]);
>> subplot(2,2,3);
>> plot(x,y1,x,y2,x,y3);
>> title('y=x.^2');
>> xlabel('x');
>> ylabel('y');
>> grid on
>> x=0:0.1:10;
>> y1=log(x);
>> y2=2*log(x);
>> y3=3*log(x);
>> axis([0 10 0 30]);
>> subplot(2,2,4);
```

```
>> plot(x,y1,x,y2,x,y3);
>> title('y=ln(x)');
>> xlabel('x');
>> ylabel('y');
>> grid on.
```

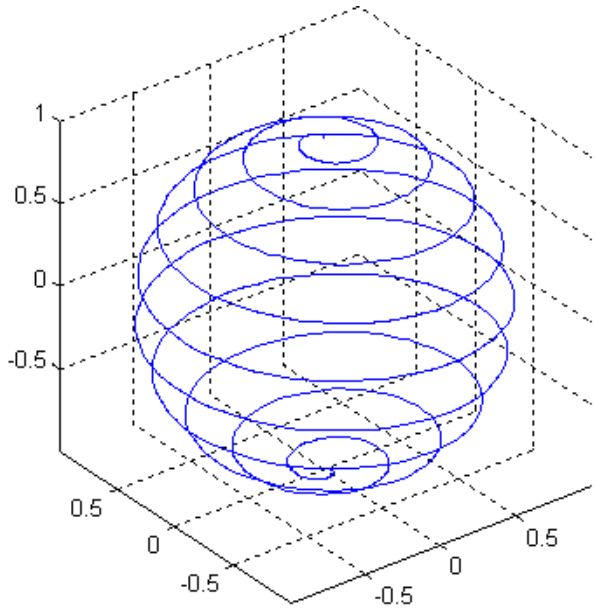



3.1 сурет – Берілген функциялардың графиктері

3.4.2 Үшөлшемді графиктерді шығару.

Нүктелерінің координаталары тек бір t параметрінің функциясымен анықталатын кеңістіктік қисықтың графигін салайық.

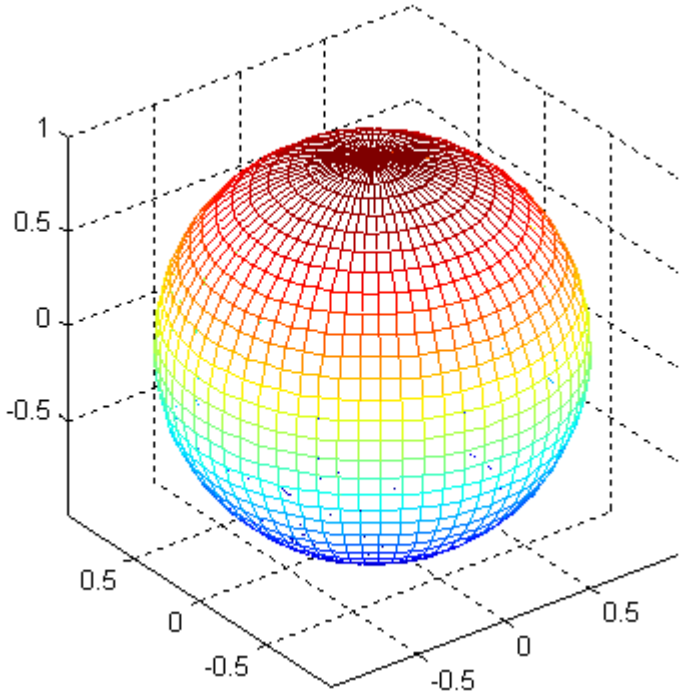
```
>> t=0:0.01:2*pi;
>> x=cos(10*t).*sin(t/2);
>> y= sin(10*t).*sin(t/2);
>> z= cos(t/2);
>> plot3(x,y.z);
>> grid on;
>> axis equal.
```




3.2 сурет – Кеңістіктік қисықтың графигі

Жазықтық графигін құрастырғанда нүктелер координаталары екі φ және θ параметрлері функцияларымен анықталады. Параметрлік қатынастар негізінде түйіндік нүктелер және осы түйіндік нүктелердегі функция мәндерінің матрицалары құрылады, осыдан кейін бет mesh() surf() функциясы көмегімен құрастырылады.

```
>> phi=0:0.1:2*pi;
>> theta=0:0.1:pi;
>> [Phi,Theta]=meshgrid(phi,theta);
>> X=cos(Phi).*sin(Theta);
>> Y=sin(Phi).*sin(Theta);
>> Z=cos(Theta);
>> mesh(X,Y,Z);
>> axis equal.
```



3.3 сурет – Бірліктік сфера бетінің графигі (шар)

### 3.3 Есептің мазмұны

1) Жұмыс мақсаты.
2) Листинг пен екі ішкі терезелі f1-f2 және бір тереземен f3-f4 графиктері (барлық графиктер 2 мен 3 көбейтілмейді).
3) Қорытынды.

### 3.4 Бақылау сұрақтары

1) Ішкі терезелі графиктер қалай тұрғызылады?
2) mesh командасы қандай қызмет атқарады?
3) Бағанды диаграмма қалай тұрғызылады?
4) grid on командасы не үшін қолданылады?
5) Осьтерді масштабтау қалай орындалады?
6) Графиканың негізгі амалдары.
7) Графиктерді безендіру үшін қандай амалдар қолданылады?
8) Графикалық амалдарды бағдарламалау кезінде қандай амалдар қолданылады?

## 4 Зертханалық жұмыс № 4. MATLAB ортасында бағдарламалау

Жұмыс мақсаты: MATLAB ортасында бағдарламалаудың негізгі элементтерін зерттеу.

### 4.1 Негізгі қалыптар

MATLAB-та бағдарлама кодының белгілі бір бөліктерін *n* рет орындау мүмкіндігі бар, бірақ олар *m*-файл түрінде қатты дискіге бір рет қана жазылады.

Машықтану кезінде көп жағдайда сценарий қолданылады – бұл кеңейтілуі *m* болатын файлға жазылған командалар жинағы. Сценарий - файлдарда қолданылатын айнымалылар, жүйемен жұмыс істеу сеансы бойы мәндері сақталатын «жұмыс кеңістігін» құрастырады.

Бірінші жолында «function» көрсеткіші бар *m*-файлдар функция деп аталады. Функция-файлдарда барлық айнымалылардың аттары файл ішінде болады, сонымен қатар тақырыпта көрсетілген айнымалылар аты жергілікті болып қабылданады, яғни процедурамен жұмыс аяқталғаннан кейін осы айнымалылардың барлық мәндері жоғалады.

Іс жүзінде *m*-файлдар ішкі бағдарламалар рөлін атқарады, бірақ олар негізгі бағдарламаның Command Window жұмыс жолынан тыс жазылады және қатты дискіде бөлек сақталады.

Диалогтық енгізу үшін input командасы қолданылады.
айнымалы = input(‘мән’)

Дисплейге деректерді шығару үшін *disp* (‘айнымалы немесе *мәтін’*) командасы қолданылады.

Шартты операторларға *if* операторы және таңдау операторы *switch* жатады.

*if* шартты операторларының құрылымдары:

1) if шарт
командалар
end

Командалар тек сол жағдайда орындалады, егер шарт ақиқат болса, ал егер шарт жалған (0) болса, онда командалар орындалмайды.

2) if шарт
команда_1
else
команда_2
end

3) if шарт_1
команда_1
elseif шарт_2
команда_2
. . .
elseif шарт_N
команда_N
else
команда_N+1
end.

Шарттардың топтамасы тексерілетін if шартты операторына қарағанда, switch таңдау операторы көмегімен switch кілттік сөзінен кейін көрсетілетін айнымалы нәтижесінің (немесе мәннің) тексерілуі жүргізіледі. Case кілттік сөзі, айнымалы нәтижелеріне байланысты орындалатын бағдарламалық блоктарды құрастыру үшін қолданылады.

Switch таңдау операторын шақырудың жалпы синтаксисі:

```
switch айнымалы
case нәтиже_1
команда_1
case нәтиже_2
команда_2
……..
case нәтиже_N
команда_N
otherwise
команда_N+1
end
```

MATLAB жүйесінде қайталау операторының екі түрі бар: арифметикалық және шартты.

Арифметикалық қайталау операторы келесідей болады:

```
for айнымалы = <бм>: <қ>: <см>
командалар
end
```

мұндағы <бм> – айнымалының бастапқы және <см> – соңғы мәні.

<қ> мәні айнымалының <бм>-нен бастап <см>-не дейінгі өсімше қадамын береді. Егер <қ> берілмесе, онда ол үндеу бойынша 1-ге тең.

Командаларды бекітілмеген сан рет қайталау үшін алғышартты while қайталау операторы қолданылады:

```
while шарт
командалар
end
```

Егер «шарт» айнымалысының нөлдік емес элементтері болса, операторлар орындалады.

Компьютердің оперативті жадында файлдар мен деректермен жұмыс істеу үшін келесі командалар қолданылады:

1) dir – берілген қалтада сақталынған файлдар жиынтығын экранға шығару;

2) what – дискіде бар m-файлдардың аттарын шығару;

3) type – экранға мәтіндік файлдың листингін шығару.

## 4.2 Жұмысты орындау тәртібі

4.2.1 Сценарийлерді құрастыру.

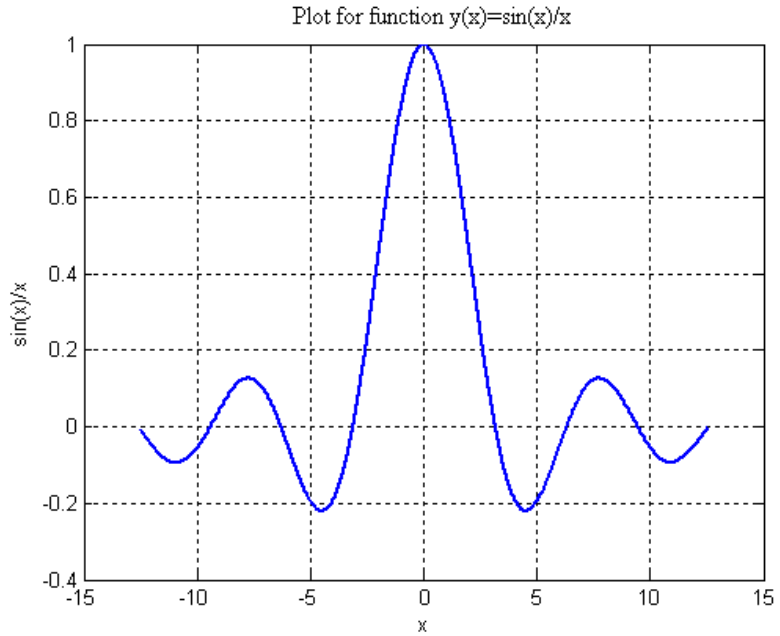
Сценарий m-файлы жасалатын Editor кірістірілген өңдегішін шақыру үшін Command Window терезесінің негізгі мәзірінде File-New-Blank M-File командасын таңдау қажет.

Editor өңдегішінде келесі командаларды жазыңыз:

```
x=-4*pi:0.01:4*pi;
y=sin(x)./x;
plot(x,y,'b-','LineWidth',2)
title('Plot for function y(x)=sin(x)/x') үлгі (4.1 суретті қара)
xlabel('x')
ylabel('sin(x)/x')
grid on
```

Файл-сценарийді SinX.m атымен сақтап, жабыңыз. Содан кейін Command Window жұмыс жолында сценарий файлының атын терген кезде, y(x)=sin(x)/x функциясының графигі пайда болады (4.1 суретті қара). sin(x)./x кейінгі нүкте x аргументінің радиан қадамы 0,01 болатын берілген x=-4*pi:0.01:4*pi арақашықтықтағы белгілі бір мәніндегі вектор-функцияның (матрица-жол) құрастырылуын білдіреді.

```
>> SinX
>>
```




4.1 сурет – Сценарий орындалғаннан кейінгі функция графигі

4.2.2 Қолданушы функциясы.

Функцияның *m*-файлы жасалатын Editor кірістірілген өңдегішін шақыру үшін Command Window терезесінің негізгі мәзірінде File-New-Function M-File командасын таңдау қажет.

Editor өңдегішінде келесі командаларды жазыңыз:

```
function z=ComplPoc
Re=input('Re=');
Im=input('Im=');
```

```
z=Re+1i*Im;
disp(['z=',num2str(z)]);
md=sqrt(Re^2+Im^2);
disp(['md=',num2str(md)]);
rd=atan2(Im,Re);
disp(['rd=',num2str(rd)]);
phi=(180/pi)*rd;
disp(['phi=',num2str(phi)]);
end
```

num2str() командасы жақша ішіндегі айнымалы мәнін мәтінге айналдырады.

Файл-функцияны ComplPoc.m атымен сақтаңыз. Содан кейін Command Window жұмыс жолында функцияның атын терген кезде, *z* комплекс санын көрсетудің алгебралық түрінен көрсеткіштік түріне ауыстырудың нәтижесі шығады (атауы да осы жерден).

```
>> ComplPoc
Re=3
Im=4
z=3+4i
md=5
rd=0.9273
phi=53.1301
```

4.2.3 Шартты операторлар.

4.2.3.1 *if* шартты операторы.

Editor өңдегішінде келесі командаларды жазыңыз:

```
function y=Select(x)
if x<-5
   y=x+5;
   disp(['y=',num2str(y)]);
elseif x<0
   y=x*(x+5);
   disp(['y=',num2str(y)]);
elseif x<7
   y=x*(x-7);
   disp(['y=',num2str(y)]);
else
   y=x-7;
   disp(['y=',num2str(y)]);
end
if y==0
   disp('bye-bye');
end
end
```

Файл-функцияны *Select.m* атымен сақтаңыз. Содан кейін *Command Window* жұмыс жолында функцияның атын *x* аргументінің кез келген мәнімен терген кезде, аргумент таңбасына байланысты *y* функциясын есептеу нәтижесі шығады.

```
>> Select(2)
y=-10
>> Select(10)
y=3
>> Select(-3)
y=-6
>> Select(-10)
y=-5
>> Select(0)
y=0
bye-bye
```

4.2.3.2 Таңдау операторы switch.

Editor өңдегішінде келесі командаларды жазыңыз:

```
function y=Perebor(x,n)
switch n
   case 1
      y=x;
   case 2
      y=x*(10-x);
   case {3,4,5}
      y=x*sin(n*x);
   otherwise
      y=1/(1+x^2);
end
end
```

Файл-функцияны *Perebor.m* атымен сақтаңыз. Содан кейін *Command Window* жұмыс жолында функцияның атын *x* аргументінің кез келген мәні және *n* айнымалысымен (екі аргумент) терген кезде, *x* және *n*-ға байланысты *y* функциясын есептеу нәтижесі шығады.

```
>> Perebor(5,1)
ans =
    5
>> Perebor(5,2)
ans =
   25
>> Perebor(5,3)
ans =
   3.2514
>> Perebor(5,4)
```

```
ans =
   4.5647
>> Perebor(5,-1)
ans =
   0.0385
```

4.2.4 Қайталау операторлары.

4.2.4.1 Арифметикалық қайталау операторы.

Editor өңдегішінде келесі командаларды жазыңыз:

```
function y=Expan(x,n)
z=0;
q=1;
for i=0:n-1
z=z+q;
q=q*x/(i+1);
end
y=z+q;
end
```

Файл-функцияны *Expan.m* атымен сақтаңыз. Содан кейін *Command Window* жұмыс жолында функцияның атын *x* аргументінің кез келген мәнімен (экспонента дәрежесі) және *n* айнымалысымен (қатар қосындысының жоғарғы шегі) терген кезде, $\exp(x) = \sum_{k=0}^{n} \frac{x^k}{k!}$ экспонентасының жуықталған мәнін есептеу нәтижесі шығады.

Егер *k*-қадамында өсімше $q_k = \frac{x^k}{k!}$ -ға тең болса, келесі k+1-қадамында $q_{k+1} = \frac{x^{k+1}}{(k+1)!} = q_k \cdot \frac{x}{k+1}$.

```
>> Expan(1,5)
ans =
   2.7167
>> Expan(1,100)
ans =
   2.7183
```

4.2.4.2 Шартты қайталау операторы.

*Editor* өңдегішінде келесі командаларды жазыңыз:

```
function s=Qvadrat(n)
sum=0;
i=1;
while i<=n
   sum=sum+i*i;
   i=i+1;
```

```
end
s=sum/n;
end
```

Файл-функцияны *Qvadrat.m* атымен сақтаңыз. Содан кейін *Command Window* жұмыс жолында функцияның атын *n* аргументінің кез келген мәнімен терген кезде, $S = \frac{1}{n}\sum_{k=1}^{n} k^2$ функциясын есептеу нәтижесі шығады.

```
>> Qvadrat(10)
ans =
  38.5000
```

### 4.3 Есептің мазмұны

1) Жұмыс мақсаты.
2) Шартты және қайталау операторлары бағдарламаларының листингі.
3) Қорытынды.

### 4.4 Бақылау сұрақтары

1) Сценарий мен функцияның айырмашылығы неде?
2) Пернетақтадан айнымалы жұмыс ортасын қалай енгізеді?
3) Шартты операторлар не үшін қажет?
4) Қайталау операторлары не үшін қажет?
5) Бағдарламалау кезінде m-файлдар қандай қызмет атқарады?
6) Таңдау операторлары не үшін қажет?
7) Matlab ортасында бағдарламалаудың негізгі элементтерін атаңыз?
8) Арифметикалық қайталау операторы дегеніміз не?

## 5 Зертханалық жұмыс № 5. MATLAB қолданушының функциялары

Жұмыс мақсаты: MATLAB жүйесінің математикалық функцияларын зерттеу және қолданушы функцияларымен жұмыс істеу бағдарламаларын құрастыру.

### 5.1 Негізгі қалыптар

Қолданушы функцияларымен жұмыс істеу кезіндегі қалыпты сандық әдістерге жатады:

- бір теңдеудің түбірлерін табу (fzero);
- теңдеулер жүйесінің түбірлерін табу (fsolve);
- полином түбірлерін табу (roots);
- бір аргумент функциясының жергілікті минимумы мен максимумын іздеу (fminbnd);
- бірнеше аргумент функциясының минимумын іздеу (fminsearch);

- анықталған интегралды есептеу (quad);
- өзіндік емес интегралды есептеу (quadgk);
- дифференциалды теңдеулерді шешу (ode45).

## 5.2 Жұмысты орындау тәртібі

5.2.1 Теңдеулерді шешуге арналған кірістірілген функциялар.

5.2.1.1 fzero() функциясы.

fzero() функциясын бірнеше әдіспен шақыруға болады. Бірінші аргумент болып *f(x)* функциясының өзіне жасалатын функцияға көрсеткіш, функция коды бар *m*-файл аты немесе *f(x)* функционалды байланысын анықтайтын, жақшаға алынған формула түрінде берілген сілтеме саналады. *fzero*() функциясының екінші аргументі болып іздеу интервалының шектері *(f(x)* функциясы әртүрлі таңбалардың мәндерін қабылдау керек немесе бастапқы жуықтау (іздеудің бастапқы нүктесі) саналады.

1. Іздеу интервалында теңдеуді шешу

```
>> x=fzero('2*sin(x)^2 + sin(x) - 1',[0 pi/2])
x=
    0.5236
>> x=fzero('2*sin(x)^2 + sin(x) - 1',[pi/2 pi])
x=
    2.6180
```

2. Теңдеуді *m*-файлға сілтеме көмегімен және бастапқы жуықтаумен шешу *f.m* атты *m*-файлдың *Editor* өңдегішінің терезесі келесідей болады:

```
function z=f(x)
z=2*sin(x)^2 + sin(x) - 1;
end
```

Командалар терезесінде *f.m* атты *m*-файлға сілтемені екі әдіспен жасауға болады:

```
>> z=fzero(@f,pi/4)
x=0.5236
>> z=fzero(' f ',3*pi/4)
x=2.6180
```

3. Анонимдік функциялар @(x)

@(x) анонимдік функцияларын қолдану шешілетін *f(x)* функциясын тікелей командалар терезесінде жариялауға және тек теңдеу түбірін ғана емес, сонымен қатар [*x, res*] командасы көмегімен функция мәнін алуға мүмкіндік береді.

```
>> F=@(x) (sin(x)+cos(x))
F=
    @(x) (sin(x)+cos(x))
>> [x,res]=fzero(F,[0 pi])
x=
```

2.3562

res=

1.1102e-016

Көріп отырғанымыздай, x=3π/4=2.3562 түбірі нүктесінде F функциясының мәні res=1.11x10$^{-16}$ жуықтап алғанда нөлге тең.

5.2.1.2 fsolve() функциясы.

Теңдеулер жүйесін шешу үшін, бірінші аргументі вектор-функцияға сілтеме болатын *fsolve*() функциясы қолданылады. Сілтеме *m*-файл-ға немесе @ анонимдік функцияға болу мүмкін, яғни ертеректе жазылған әдістермен бірдей.

Теңдеулер жүйесін шешейік:

$$x_1^2 + x_2^2 - 25 = 0$$

$$(x_1 + 1)x_2 - 16 = 0$$

```
>> opts=optimset('Display','off');
>> F=@(x) [(x(1)^2 + x(2)^2 - 25),((x(1) + 1)*x(2) - 16)];
>> z=fsolve(F,[0 1],opts)
z=
    3.0000    4.0000
```

*Optimset ('Display','off')* командасы экранға (қажет емес) *Matlab* мәтіндік ақпаратының шығарылуын тоқтатады. Іздеудің бастапқы нүктелері $x_1$=0 және $x_2$=1-ге тең.

5.2.1.3 *roots*() функциясы.

Автоматты басқару теориясында жүйенің беріліс функциясының полиномдарының түбірлерін табу қажеттігі көп туады. Ол үшін, полином коэффициенттері тізімін алдын ала құрып, *roots*() функциясын қолданса болады.

6 дәрежелі полином берілген:

$$P(x) = x^6 - 10x^5 + 16x^4 - x^2 + 10x - 16.$$

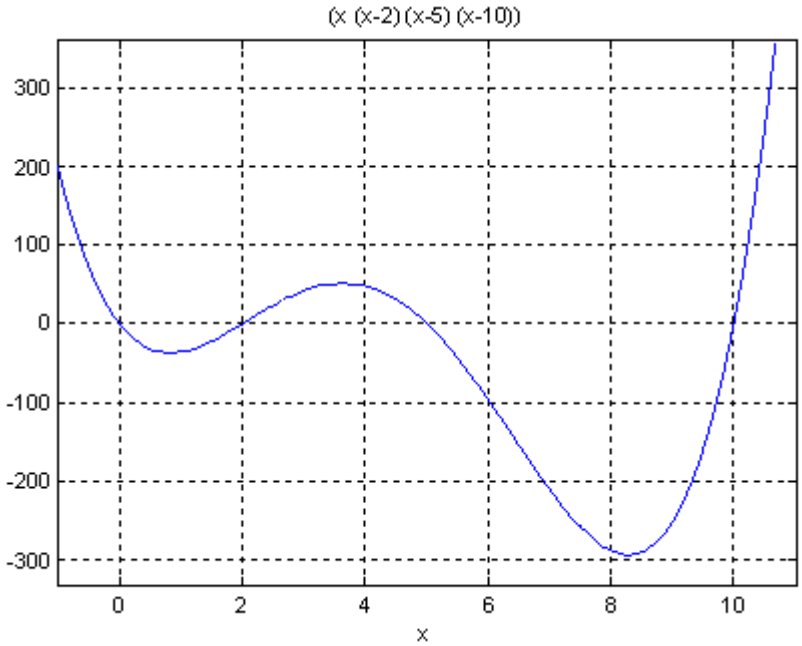
```
>> a=[1,-10,16,0,-1,10,-16]
a=
    1   -10    16     0    -1    10   -16
>> roots(a)
ans=
    8.0000
    2.0000
   -1.0000
   -0.0000 + 1.0000i
   -0.0000 - 1.0000i
    1.0000
```

5.2.1.4 *fminbnd*() функциясы.

Шекті интервалда бір аргументтің скалярлық функциясының жергілікті минимумын іздеу үшін *fminbnd*() функциясы қолданылады.

f(x)=x(x - 2)(x - 5)(x - 10) функционалдық байланысы 2 минимум және 2 максимумға ие. *ezplot(f,[-1 11])* графикті тез тұрғызу функциясында бірінші аргумент болып функцияға көрсеткіш саналады, ал екіншісі – аргументтің өзгеру диапазоны болады.

```
>> f=@(x) (x.*(x - 2).*(x - 5).*(x - 10);
>> ezplot(f,[-1 11]);
>> grid on;
>> [x y]=fminbnd(f,-1,11);
x=8.2874;
y=-293.3580;
>> [x y]=fminbnd(f,-1,2);
x=0.8306;
y=-37.1338.
```




5.1 сурет – Экстремумды іздеу функциясының графигі

*y=f(x)* функциясының жергілікті максимумын табу үшін *y=-f(x)*-тің жергілікті минимумын табу қажет. *fminbnd(@(x)(-f(x),2,6)* функциясының бірінші аргументі ретінде *x* аргументіне әсері *f(x)* мәнін есептеуден құралатын, мұндағы *f* бастапқы функцияға көрсеткіш болатын *@(x) (-f(x))* анонимдік функция көрсетілген.

```
>> f=@(x) (x.*(x - 2).*(x - 5).*(x - 10));
>> ezplot(f,[-1 11]);
>> grid on;
>> [x y]=fminbnd(@(x) (-f(x),2,6)
x=3.6320
y=-51.6363
```

5.2.1.5 *fminsearch*() функциясы.

Бірнеше аргумент функциясының минимумын табу үшін *fminsearch*() функциясы қолданылады. Функция аргументтері болып зерттелетін функцияға көрсеткіш және берілген функцияның бастапқы жуықтауларының тізімі саналады.

$f(x,y) = x^2 - 2xy + 2y^2 - 4x$ функциясының минимумын анықтаймыз.

```
>> f=@(x) (x(1)^2 - 2*x(1)*x(2) + 2*x(2)^2 - 4*x(1));
>> [x z]=fminsearch(f,[3,1])
x=
    4.0000    2.0000
z=-8.0000
```

Функция минимум x=4 және y=2 нүктелерінде, ал мәні f=-8.

5.2.2 Анықталған интегралдарды есептеу.

5.2.2.1 *quad*() функциясы.

quad() функциясы Симпсон квадратуралары әдісімен анықталған интегралды есептеу үшін қолданылады, оның бірінші аргументі ішкі интегралды функцияға сілтеме болады (немесе сәйкес *m*-файлдың аты), ал екінші және үшінші – интегралдау шектері болады.

$\int_0^{\pi/4} tg(x)dx = \frac{\ln(2)}{2}$ интегралын табамыз.

```
>> quad(@tan,0,pi/4)
ans=0.3466
>> log(2)/2
ans=0.3466
```

5.2.2.2 *quadgk*() функциясы.

*quadgk*() функциясы өзіндік емес интегралды Гаусс-Кронрод квадратуралар әдісімен есептеу үшін қолданылады (шектері 0-ден ∞-ке дейін). *quad*() және *quadl*() функциялары бұндай интегралды есептей алмайды.

$\int_0^{\infty} \exp(x)dx = \frac{\sqrt{\pi}}{2}$ интегралын табамыз (Пуассон интегралы).

```
>> f=@(x) (exp(-x.^2));
>> quadgk(f,0,Inf)
ans=0.8862
>> sqrt(pi)/2
ans=0.8862
```

5.2.3 Дифференциалды теңдеулерді шешу.

Сандық түрде дифференциалдық теңдеулер Коши есебі түрінде шешіледі – бастапқы мәндері беріледі.

*ode45*() функциясы Рунге-Кутты әдісімен 4 және 5 ретті дифференциалдық теңдеулерді шешу үшін қолданылады, оның бірінші аргументі болып шешілетін функцияға көрсеткіш саналады, екінші болып *x* аргументтерінің мәндер тізімі, ал үшінші аргумент функцияның бастапқы мәндерін анықтайды.

Бастапқы мәні y(1)=1 болатын $y^1 = y^2 - 2/x^2$ дифференциалдық теңдеуді шешеміз.

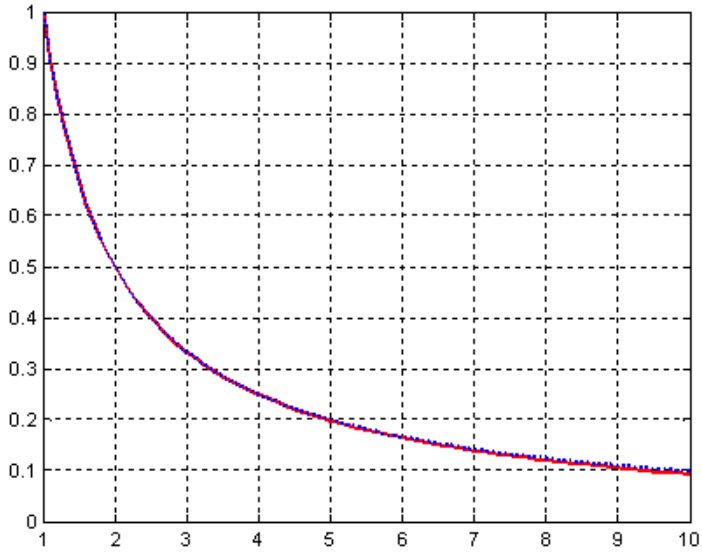
Жалпы шешім келесі түрде болады $y(x) = \dfrac{3}{(1+Cx^2)x} - \dfrac{2}{x}$,

мұндағы C - константа, бастапқы шарттардан анықталады.

y(1)=1 бастапқы шартын қойып, $1 = \dfrac{3}{1+C} - 2$ немесе C=0 аламыз.

Сонда шешім келесі түрде болады y(x)=1/x.

```
>> f=@(x,y) (y.^2 - 2./x.^2)
>> [x,y]=ode45(f,1:0.1:10,1)
>> plot(x,y,'r-','LineWidth',2)
>> hold on
>> plot(x,1./x,'b:','LineWidth',2)
>> grid on
```




5.2 сурет – Дифференциалдық теңдеудің шешімінің графигі

Бірінші график ode45 командасынан кейінгі шешімді қалыңдығы 2 қызыл сызықпен көрсетеді, ал екінші график y=1/x көк пунктир сызықпен көрсетеді.

### 5.3 Есептің мазмұны

1) Жұмыс мақсаты.

2) f(x) = 0 теңдеуінің түбірлерін табу үшін арналған бағдарлама листингі.

3) Бір айнымалы функциясының минимумын табу үшін арналған бағдарлама листингі;

4) Бірнеше айнымалы функциясының минимумын табу үшін арналған бағдарлама листингі.

5) Анықталған интегралды есептеуге арналған бағдарлама листингі.

6) Дифференциалдық теңдеулерді шешуге арналған бағдарлама листингі.

**5.4 Бақылау сұрақтары**

1) fzero функциясы не үшін арналған?
2) fsolve функциясы не үшін арналған?
3) roots функциясы не үшін арналған?
4) fminbnd және fminsearch функцияларының айырмашылықтары неде?
5) quad және quadgk функцияларының айырмашылықтары неде?
6) ode45 45 функциясы не үшін арналған?
7) Іздеу интервалында теңдеуді шешу қалай жүзеге асырылады?
8) @(x) анонимдік функциялары қандай қызмет атқарады?

## Әдебиеттер тізімі

1 Васильев А.Н. MATLAB. Практический подход. - СПб.: Наука и Техника, 2012. - 448 с.

2 Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем. − М.: Радио и связь, 1988. − 560 с.

3 Дьяконов В.П. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. – М.: СОЛОН-Пресс, 2002. – 768 с.34. Дьяконов В.П. VisSim+Mathcad+MATLAB. Визуальное математическое моделирование. – М.: СОЛОН-Пресс, 2004. – 384 с.

4 Дьяконов В. П. Справочник по применению системы PC MATLAB. - М.: Физматлит, 2004. - 112 с.

5 Хайнеман Р. Моделирование работы электронных схем. - М.: ДМК Пресс, 2005. - 336 с.

## Мазмұны

Кіріспе 3
1 Зертханалық жұмыс №1 3
2 Зертханалық жұмыс №2 8
3 Зертханалық жұмыс №3 14
4 Зертханалық жұмыс №4 19
5 Зертханалық жұмыс №5 26
Әдебиеттер тізімі 33

2013 ж. жиынтық жоспары, реті 160

Самал Бексултановна Абдрешова

АСПАП ЖАСАУДАҒЫ КОМПЬЮТЕРЛІК ТЕХНОЛОГИЯЛАР

5В071600 – Приборлар жасау мамандығы студенттері үшін
зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Редактор З.С. Советова
Стандарттау бойынша маман Н.Қ. Молдабекова

Басуға қол қойылды _____ Пішіні 60х84/16
Таралымы 50 дана. № 1 типографиялық қағаз
Көлемі 2,1 оқу-бас.ә. Тапсырыс_.Бағасы 1050 тг.

Алматы энергетика және байланыс университеті
коммерциялық емес акционерлік қоғамының
көшірме-көбейткіш бюросы
050013 Алматы, Байтұрсынұлы көшесі, 126