

**O'ZBEKISTON RESPUBLIKASI AXBOROT
TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT
AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

A.H.NISHANOV, A.T.RAHMANOV, M.X.AKBAROVA

AMALIY DASTURIY PAKETLAR

O'QUV QO'LLANMA

Toshkent 2019

UO'K: 004.9(075.8)
KBK: 32.973

A.H.Nishanov, A.T.Rahmanov, M.X.Akbarova. Amaliy dasturiy paketlar. O'quv qo'llanma. – T.: «Aloqachi» 2019. – 224 bet.

ISBN 978–9943–5806–4–0

O'quv qo'llanma ilmiy–texnik hisoblar va ishlab chiqarish tizimlarining modellashtirish masalalarini o'rganishda keng imkoniyatlarga ega bo'lgan amaliy dasturiy paketlar tizimlarining ajralmas qismi bo'lgan Matlab, Maple, Mathcad va boshqa kompyuter tizimlari haqida asosiy bazaviy ma'lumotlardan iborat.

Qo'llanmada Matlabda ma'lumotlarni kiritish va tashkil etish, qayta ishlash, ular ustida amallar, funksiyalar va operatorlarning tavsiflari, ikki va uch o'lchovli grafiklar, algebraik tenglamalar va ularning sistemalari, optimallashtirish masalalari va ularni yechish uchun amaliy vositalar hamda vizualizatsiya masalalariga asosiy e'tibor qaratilgan, mavzular misollar yordamida illyustratsiya qilingan.

Talabalar, professor–o'qituvchilar, katta ilmiy xodim-izlanuvchilar va mustaqil o'rganuvchilarning keng ommasiga mo'ljallangan.

O'quv qo'llanma Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti ilmiy uslubiy kengashining qaroriga asosan chop etildi.

UO'K: 004.9(075.8)
KBK: 32.973

Taqrizchilar: D.T.Muhamediyeva;
M.X.Hakimov.

Ma'sul muharrir: K.F.Kerimov.

ISBN 978–9943–5806–4–0

© «Aloqachi» nashriyoti 2019.

KIRISH

Bugungi kunda fan-texnika olamida murakkab bo'lgan masalalarni yechish uchun turli xil dasturlash tillari va vositalardan foydalaniladi. Kompyuter texnologiyalarining keng amaliyotga qollanishi dasturlashning rivojlanishi bilan uzviy holda yuz beradi. Ilmiy-texnika va texnologiyalarning rivojlanishi oqibatida murakkab masalalarning matematik hamda dasturiy ta'minotini ishlab chiqishga talab kuchayadi. Hozirgi davrga kelib kompyuter va kompyuter texnologiyalaridan foydalanuvchilar toifasi shunchalik xilma-xilki, ularning barchasidan yuqori darajadagi dasturlash tillarini bilishni talab qilish imkoni yo'q.

Bunday toifadagi foydalanuvchilar uchun, nisbatan oson qo'llaniladigan dasturiy vositalar –matematik *amaliy dasturlar paketlari* (ADP) mavjud. Xususan, bunday tizimlarga kompyuter algebrasining keng imkoniyatli paketlari - *Mathematica, Maple, Matlab, MathCAD, Mercury, Statistica, Derive* va boshqalarni qo'shish mumkin. Bu tizimlarda hisoblash jarayonlarida bir qator doimiy takrorlanuvchi standart jarayonlar alohida “paket” deb ataluvchi maxsus dasturlar tarkibiga kiritiladi. Dasturlar paketi o'z navbatida ob'ektlilikni vujudga keltiradi.

Amaliy masalalar turli paketlarga bo'linib, “kompyuter algebrasi” deb ataluvchi bir nechta dasturiy ta'minotlar tarkibiga kiritilgan bo'ladi. Ulardan, *Mathematica va Maple* professional matematiklar uchun mo'ljallangan bo'lib, imkoniyatlarning boyligi, ishlatishda murakkabligi bilan ajralib turadi. Matlab dasturi matrisalar bilan ishlashga va signallarni avtomatik boshqarish hamda qayta ishlashga mo'ljallangan bolib, ikki va uch o'lchovli grafiklarni vizualizatsiyalashda Maple imkoniyatlarini o'zida mukammallashtirgan tizimlardan biri hisoblanadi. MathCAD va Derive esa sodda qo'llanilishga mo'ljallangan tizimlardan bolib, ko'p sonli foydalanuvchilarning talablarini qondirishni ta'minlaydi.

Ushbu o'quv qo'llanma “Kompyuter inginiring”, “Dasturiy inginiring” va boshqa dasturlash tizimlari bilan bog'liq bo'lgan ta'lim yo'nalishlarida o'tiladigan asosiy fanlardan biri bo'lgan “Amaliy dasturiy paketlar” fanini mukammal o'rganishga bag'ishlangan bo'lib, unda Matlab tizimining eng sodda tushunchalaridan boshlab turli xil amaliy masalalarni yechishga mo'ljallangan ob'ektlilikni ochib berilgan.

Ma'lumki, juda ko'p amaliy masalalarni yechish uchun uning ma'lum ko'rinishdagi matematik modeli ishlab chiqiladi va uni yechish mutaxassis tomonidan hal etiladi. Buning uchun quyidagi masalalarni ketma-ket yechish lozim bo'ladi:

1. Masalani ifodalovchi lingvistik model yordamida berilgan boshlang'ich qiymatlar va qiymatlari qidirilayotgan miqdorlar o'rganilib, masalani yechish uchun zarur bo'lgan parametrlar majmuasini aniqlash.

2. Masalaning mohiyatidan kelib chiqib, matematik va boshqa qonuniyatlardan foydalangan holda, parametrlar orasida munosabatlar o'rnatish, ya'ni qo'yilgan masalaning matematik modelini ishlab chiqish.

3. Matematik modelni yechish uchun biror hisoblash usulini tanlash va unga asoslab algoritm va dasturiy ta'minot ishlab chiqish.

4. Kompyuterda tajribalar o'tkazib, modelning adekvatligini tekshirish.

Yuqorida keltirilgan jarayon, modellashtirish yordamida amaliy masalalarni yechish hisoblanadi. Har bir masala ma'lum bir sinfga tegishli bo'lgani uchun, bunday sinf masalalarini yechishga mo'ljallangan dasturiy vositalar, amaliy dasturlar paketlari ishlab chiqish juda muhim hisoblanadi. Ana shunday dasturlar paketlari yuqorida keltirilgan *Mathematica, Maple, Matlab, MathCAD, Mercury, Statistika, Derive* va boshqa tizimlarda ishlab chiqilgan va bu jarayon davom etmoqda.

Foydalanilayotgan tizim matematik paketlarini shartli ravishda ikki guruhga ajratish mumkin: belgili(simvulli) matematika dasturlari va masalalarni sonli yechishga qaratilgan dasturlar. Yuqorida keltirilgan tizimlardan *Statistica, Derive* kabi paketlar matematik masalalarni sonli usullar bilan yechishga mo'ljallangan.

Paketlardan ayrimlari ikki guruh funksiyalarini ham bajara oladi. Hozirgi vaqtda bunday paketlardan etakchilari *Matlab, MathCAD, Mathematics, Maple* lar hisoblanadi. Bu paketlar simvolik va analitik almashtirishlar hamda turli sonli usullarni qo'llash bo'yicha keng imkoniyatlarga ega. Alohida ta'kidlash kerakki, ular ilmiy masalalarni echishga ham moslashtirilgan bo'lib, ilmiy-tadqiqot o'tkazish uchun juda qulay vosita hisoblanadi. Shu sababli bu paketlar ta'lim tizimida va ilmiy sohada keng ommalashgan. O'qitishda kerakli paketni tanlashdan avval uning imkoniyatlarini baholash zarur bo'ladi. Matematik dasturiy tizimlarning eng soddasi va foydalanishga qulayi hisoblangan *MathCAD* va *Matlab* tizimlari haqida qisqacha to'xtalib o'tamiz.

MathCAD haqida gapiradigan bo'lsak, u xar-xil soha masalalarini modellashtirib, matematik usullar yordamida yechish uchun mo'ljallangan integrallashgan muhit bo'lib, quyidagi funksional komponentlardan iborat:

- koordinatsiyalashgan va qulay menyular tizimi, kontekst menyusu;
- vositalar paneli majmuasi;
- matn muharriri;

- simvollar bilan ishlovchi formulalar tahrirlagichi;
- grafik tahrirlagich, xususan ikki, uch ulchovli grafiklarni(sirtlarni) chizish va o'rganish imkoniyatini beradi;
- sonli va simvolli hisoblashlar imkoniyatini beruvchi hisoblash tizimi;
- maxsus matematik belgilarni va formulalarni kiritish uchun mo'ljallangan shablonlar majmuasi;
- matematik ifodalarni sintaksis tahrirlashga ko'mak beruvchi yordam tizimi.

MathCAD menyusi ierarxik tuzilishda bo'lib, bosh menyu ya'ni,gorizontal menyu punktlariga bog'langan osiluvchi vertikal menyu va uning qo'shimcha menyulari, qalqib chiquvchi menyu, kontekst menyulardan iboratdir.

MathCAD dasturiy tizimi Math Soft Inc. firmasi tomonidan kompakt disklarda chiqariladi. Uni standart usullar bilan installyasiya qilinadi. MathCAD dasturi o'rnatilgach, Windows OSning bosh menyusida qayd etiladi. *Fayl, pravka,vid, vstavka, format, okno, pomoh* menyulari har qanday Windows dasturlarining menyulari uchun standart vazifalarni bajaradi.

MathCAD paketi kuchli matematik apparatga ega. U sozlangan matematik funksiyalar bilan bir qatorda, matritsalar bilan ishlash, trigonometriya, oddiy differensial tenglamalarni sonli echish, ayrim statistic algoritmlar, chiziqli va chiziqli bo'lmagan tenglamalar sistemasini echish hamda boshqa matematik apparatlarni o'z ichiga oladi. Paket xujjatining har bir sahifasida masalaning echimi, matnlar, matematik ifodalar, ikki va uch o'lchovli grafiklar, hosil qilingan va Windows- ilovada mavjud chizmalardan iborat izohlar bilan berilishi, bajarilgan ishlar haqida paket ichida to'liq ma'lumotga ega bo'lish imkoniyatini beradi. Paketning afzalliklaridan yana biri, unda masalalarning echimlari ko'rsatilgan qulay ma'lumotlar tizimi hamda asosiy matematik, fizik formulalar va o'zgarmaslar bo'yicha ma'lumotnoma mavjud. Bunday hujjatlashtirish paketni juda ko'p yo'nalishdagi ilmiy-texnik ma'lumotlar bilan to'ldirish imkonini beradi. Bu dasturlarning har biri o'z kamchilik va yutuqlari bilan alohida o'rganib chiqishga arziydi.

Jadal sur'atlar bilan rivojlanib borayotgan kompyuterlashgan matematik tizimlar (KMT), ayniqsa, sonli hisoblashlarga yo'naltirilgan tizimlar orasida Matlab matritsali matematik tizimi alohida ajralib turadi. Matlab tizimini tashkil qiluvchi paketlar sonining bisyorligi uning juda ko'plab soha masalalarini hal qilishga joriy etish imkoniyatini beradi.

Matlab dasturi 1970-yillar oxirida Kliv Moler (Cleve Moler) tomonidan sodda hisoblash jarayonlarini bajarish uchun yaratilgan. U asosan 3-avlod EHM larida ishlash uchun mo'ljallangan edi. 1980-yillar o'rtalariga kelib Little Mathworks kompaniyasi xodimi injener Djon Litl (John N. Little) tomonidan Matlabning 4-avlod EHMlariga mo'ljallangan versiyasi ishlab chiqildi. Bu 2-versiya boshqarish tizimini modellashtirish uchun yaratilgan bo'lsa-da, tez orada boshqa ilmiy va injenerlik sohalarida ommalashib ketdi. Ushbu versiyaning birinchi versiya bilan o'xshash jihatlari ko'p bo'lib, bir nechta matematik paketlari bilan farqlanib turadi.

Bugungi kunda zamonaviy kompyuterlarda Matlabning 4, 5, 6 va 7-versiyalarini ko'rish mumkin. 4-versiyada Matlabning dastlabki versiyalari xususiyatlari saqlanib qolgan. 5-versiya tarkibida ilk bor grafik muhit ishchi stoli yaratilgan. Shu bilan birga Matlabning 5-versiyasida 16 bit va 24 bitli RGB ranglari bilan ishlash, yuqori o'lchamli matrisalar bilan ishlash, fazoviy figuralarni chizishda faqat sonlar bilangina emas, balki matematik formulalar bilan ishlash imkoniyatlari kiritilgan. Matlabning 6-versiyasi bu dastur rivojining eng asosiy cho'qqilaridan biri hisoblanadi. Bunda 5-versiyaga qo'shimcha ravishda boshqa dasturlash tillariga eksport va import operatsiyasi murakkab bo'lgan jarayonlar: matematik modelni grafika bilan ishlash, boshqa kompyuter algebrasi dasturlari bilan moslasha olish jarayonlari qo'shimcha sifatida kiritilgan.

Matlab–matematik va ilmiy-texnik hisoblashlarni amalga oshirishga mo'ljallangan eng qadimiy, uzoq vaqtlar davomida ishlab chiqilgan va tekshirilgan, avtomatlashtirilgan tizimlardan biri bo'lib, u matritsa va matritsaviy amallarning kengaytirilgan talqini ustiga qurilgan. Mazkur tushuncha uning nomida ham o'z aksini topgan: Matlab – matrix laboratory- matritsali laboratoriyani anglatadi. Ma'lumki, juda ko'plab dasturlar va ular ustida amallar bajarish sikllar orqali amalga oshiriladi. Bu esa dasturning ishlashini sekinlashtiradi va bazi-bir amallarni bajarishni dasturlash tillarida ko'p o'lchamli, xususan, ikki o'lchamli, yani matritsalarini e'lon qilishni murakkablashtiradi. Matlab da asosiy ob'ekt sifatida matritsalaridan foydalanish sikllar sonini keskin kamaytiradi.

Matlab tizimini yaratishdagi asosiy maqsadlardan biri bo'lib texnik va matematik hisoblashlarga yo'naltirilgan, foydalanuvchi uchun qulay va sonli usullarni amalga oshirish uchun taklif etib kelinayotgan an'anaviy dasturlash tillari imkoniyatlaridan ustunroq dasturlash tilini yaratish hisoblandi. Mazkur tizimni yaratishda hisoblashlar tezligini oshirishga hamda tizimning turli xil masalalarini hal qilishga moslashuvchanligiga qat'iy e'tibor qaratilgan.

Matlab tizimi dasturlashning uchta asosiy kontsepsiyasini amalga oshiradi:

- a) Modullarni, ya'ni protsedura va funksiyalarni yaratishga asoslangan protseduraviy modulli dasturlash;
- b) Ob'ektga yo'naltirilgan dasturlash (ayniqsa tizimning grafikli vositalarini joriy qilishda ahamiyatli);
- c) Foydalanuvchining grafikli interfeysi GUI (Graphics User Interface) vositalarini yaratishga mo'ljallangan vizual yo'naltirilgan dasturlash.

Umuman olganda, Matlab dasturlash tili interpretatorlar sinfiga kiradi. Demak, bundan kelib chiqadiki, tizimning har bir buyrug'i "nomi" (identifikatori) bo'yicha aniqlanadi va zudlik bilan joriy etiladi. Bu esa ixtiyoriy dasturiy kodni qism-qism bo'yicha tekshirishni osonlashtiradi.

Tizimning asosiy xususiyatlaridan biri uning ochiqligi va kengaytirish imkoniyati mavjudligidir. Tizimning juda ko'plab buyruq va funksiyalari matnli formatdagi m-fayl (.m kengaytmasi) va C/C++ fayllari ko'rinishida bo'lib, barcha fayllarni modifikatsiya qilish mumkin.

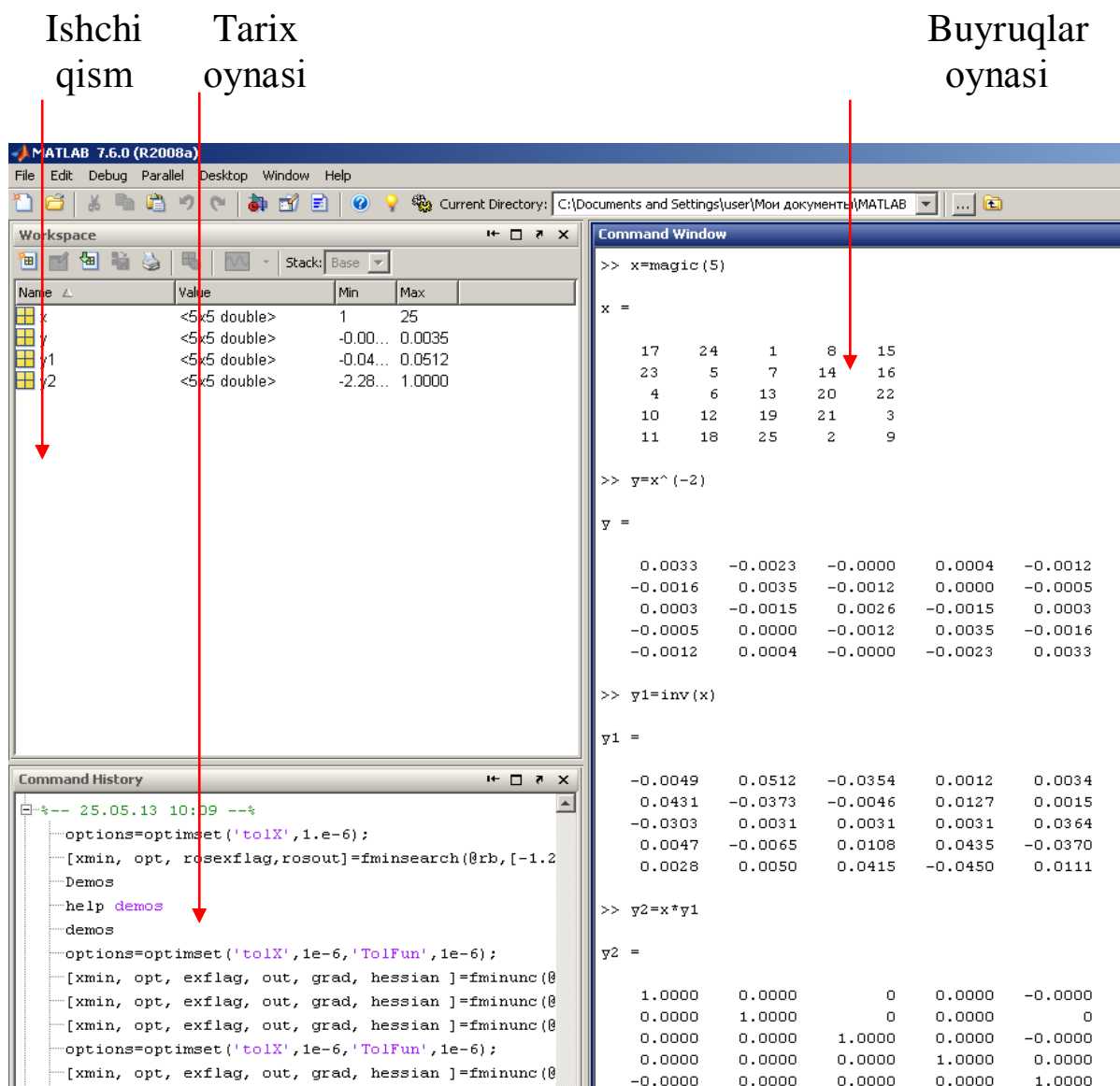
Ta'kidlash joizki, amaliy matematik dasturlar paketi bo'lgan Matlab tizimi neyron to'ri, elektrotexnik qurilmalarni modellashtirish, murakkab matematik masalalarni yechish, fizik jarayonlarni kompyuterda modellashtirish kabi ko'plab sohalarda qo'llash uchun yaratilgan.

Ingliz tilidagi intellektual mahsulot bo'lgan Matlab tizimi hozirgi kunda ilmiy – texnikaviy hisoblashlar uchun mukammal va keng ommalashgan tizim bo'lgani sababli, uni o'rganish va ayniqsa, matematika, fizika, amaliy matematika, dasturlash asoslari kabi fanlarini o'qitish jarayonida qo'llash, tabiiyki, ta'lim samarasini yanada oshiradi. Bu maqsadni amalga oshirish esa o'zbek tilida kitoblar, o'quv qo'llanmalar yaratish zaruriyatini yuzaga chiqarmoqda. Taqdim etilayotgan o'quv qo'llanma bu borada kichik bir qadam bo'ladi, deb umid qilamiz.

1. MALAB ishchi stoli va asosiy obyektlari

1.1. Matlab ishchi stoli

Matlab dasturiy ta'minotini o'rnatish jarayoni boshqa dasturlardan farqlanmaydi. Matlab dasturi ishga tushirilganda asosan 3 ta oyna ko'rinadi:



1.1-rasm. Matlab tizimining ishga tushgandan keyingi ishchi oynasi.

Umuman esa quyidagi oynalar mavjud:

1. Buyruqlar oynasi (Command Window);
2. Brouzerning ishchi qismi (Workspace Browser);
3. Massiv muharriri (Array Editor);
4. Buyruqlar tarixi oynasi (Command History);
5. Ayni vaqtdagi katalog brouzeri (Current Directory Browser);

6. Start tugmasi (Start);
7. Brouzer so'rovnomasi (Help Browser);
8. Muharrir (Editor/Debugger);
9. Sharhlovchi(Profiler).

Buyruqlar oynasi Matlab da barcha buyruqlarni, paketlarni va kutubxonalarini e'lon qilish oynasi hisoblanadi.

O'zgaruvchilar oynasi dastur tarkibida e'lon qilingan o'zgaruvchilarni daraxt ko'rinishida ifodalab boradi.

Buyruqlar tarixi oynasida esa dasturda bajarilayotgan buyruqlar ketma-ketligi saqlanib qoladi.

Matlab da seans ishi tushunchasi sessiya (session) deb yuritiladi, yani foydalanuvchi ayni vaqtda foydalanayotgan xujjat – bu sessiyadir. Unda kiritish-chiqarish satrlari va xatoliklar haqida axborot joylashgan bo'ladi. MATLAB sessiyasiga kiruvchi barcha o'zgaruvchi va funksiyalar qiymatlari xotiraning ishchi qismida joylashgan bo'ladi. Save (saqlash) komandasi yordamida ularni (Matlab.mat) faylida saqlash mumkin. Load (yuklash) komandasi esa ma'lumotlarni diskdan ishchi sohaga kiritish imkonini beradi. Diary (kundalik) komandasi orqali ma'lumotlarni ayrim qismlarini kundalik ko'rinishida saqlash mumkin.

Buyruqlar oynasini boshqarish komandalaridan eng muhimlarini keltiramiz:

- **clc** – ekranni tozalaydi va kursorni bo'sh ekranning yuqori chap qismiga joylashtiradi;

- **home** – kursorni ekranning yuqori chap qismiga qaytaradi.

Ma'lumki, o'zgaruvchilar kompyuter xotirasida, yani ishchi soha (workspace) da ma'lum bir joy egallaydi. Ishchi sohani keraksiz o'zgaruvchilardan tozalash uchun

“clear” funksiyasining turli xil ko'rinishlaridan foydalaniladi:

-**clear** - barcha aniqlangan o'zgaruvchilarni yo'qotish;

-**clear x** - aniqlangan x o'zgaruvchini yo'qotish;

-**clear a b c** - aniqlangan a b c o'zgaruvchilarni yo'qotish.

Matlab tizimiga kiritilgan o'zgaruvchilar haqida ma'lumot beruvchi komandalar ham mavjud. Ulardan biri “ who” va “ whos “ komandalaridir:

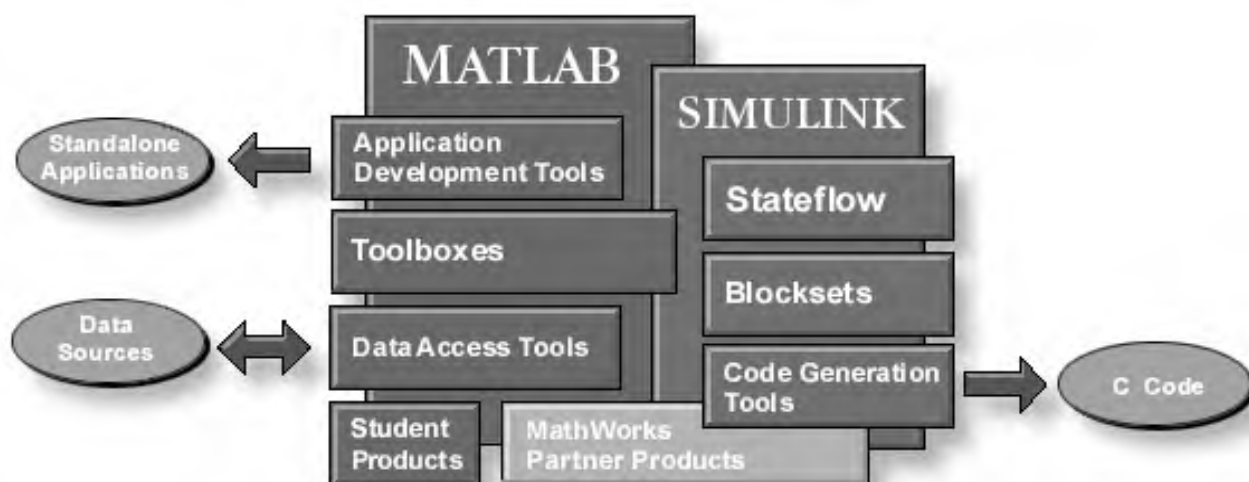
- **who** - Matlab tizimida foydalanilayotgan o'zgaruvchilar ro'yxatini hosil qiladi va ekranga chiqaradi;

-**whos** – xuddi who kabi, yana o'zgaruvchilarning o'lchovini ham chiqaradi.

1.2. Tizim kengaytmasi. Yordam tizimi

Matlab dasturchilarga quyidagi sohalaridagi paketlar kengaytmasini taqdim etadi: harbiy sanoat majmualari, energetika, aerokosmik va avtomobil qurilishi va boshqalar. Ammo shular ichidan turli tizim va qurilmalarni blokli imitatsion modelini qurish imkonini beruvchi Simulink paketi eng mashhuriga aylandi.

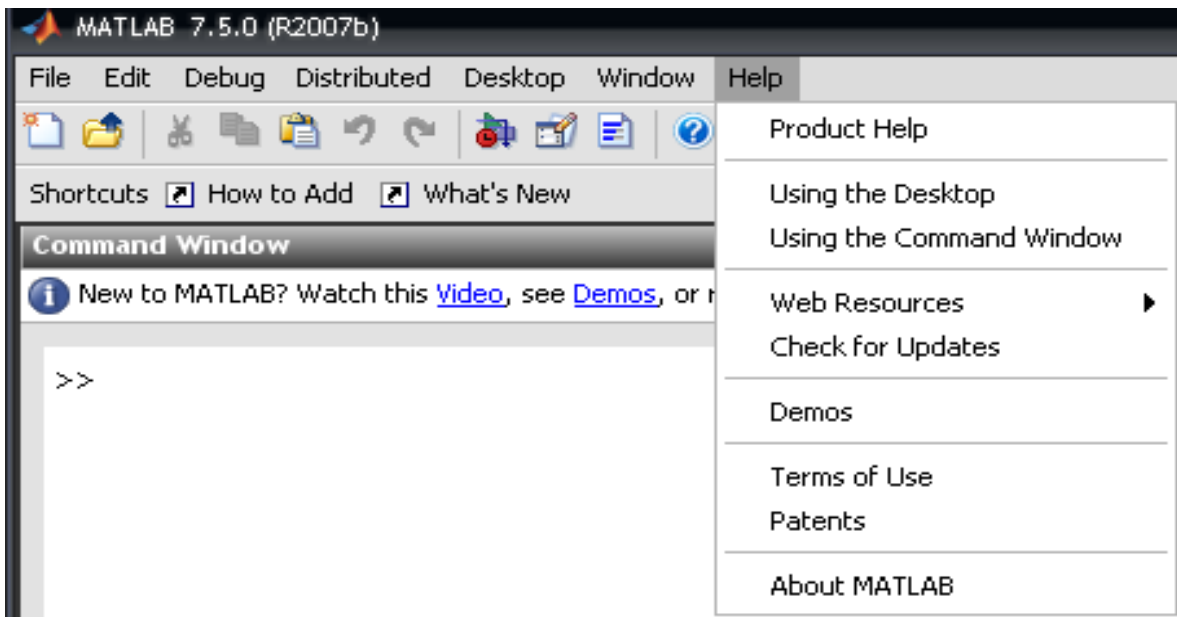
Matlab+Simulink tipik kompleksi katta hajmdagi Matlab paketlar instrumentlar “qutisi” Toolboxes va vizual-mo’ljallangan blokli imitatsion modellashtirgan Simulink dinamik tizimini imkoniyatlarini kengaytiruvchi Blocksets dan iborat. Simulink paketi Matlab bilan birga o’rnatiladi.



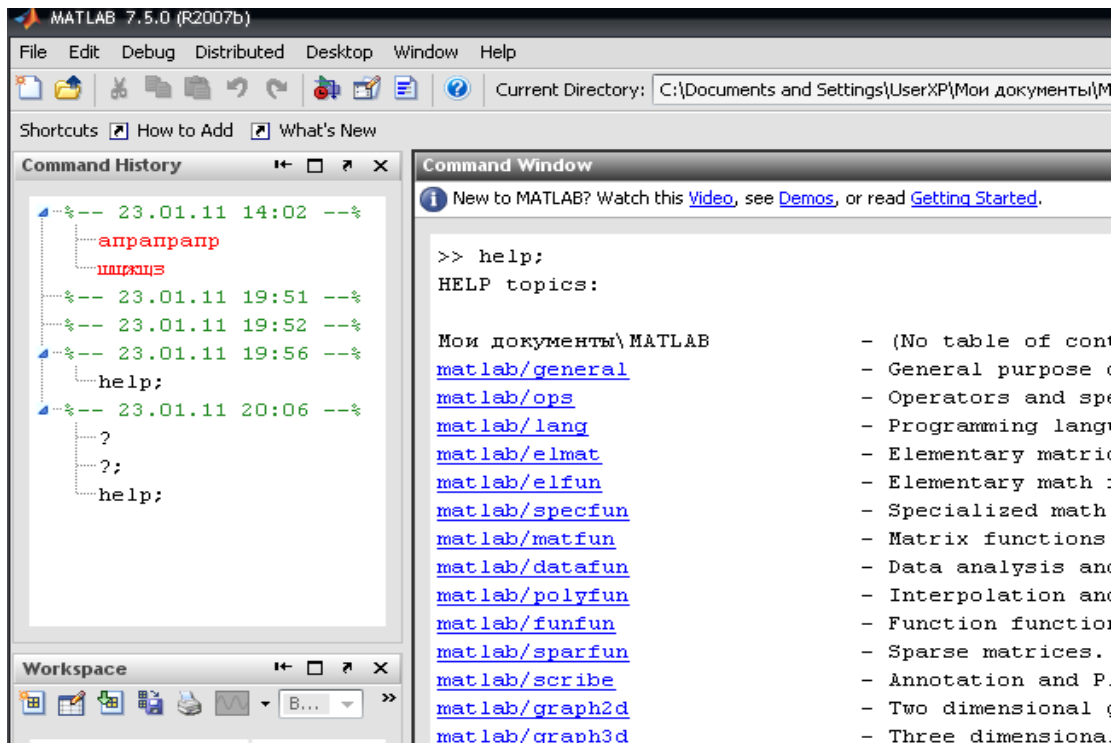
1.1 - rasm. Matlab + Simulink tizimi tuzilishi

Simulink paketi to’q’risida keyinroq batafsil ma’lumot beramiz.

Yordam tizimidan `>> help` buyruq’i orqali yoki menu panelining help bo’limiga murojaat orqali foydalanish mumkin. Help bo’limi Matlab so’rovnoma qismi va Matlab dasturi ishlab chiqarilishi haqida ma’lumot beradi.



1.2 - rasm. help buyrug'idan foydalanish.



1.3 - rasm. help buyrug'idan foydalanish.

Help komandasi berilgandan keyin ekranda <Matlab / <bo'lim> formatida yordam faylining mundarijasi chiqadi. Kerakli bo'limni tanlab, help <bo'lim> komandasi kiritiladi. Shundan keyin ekrandagi shu bo'limdagi funksiya, o'zgaruvchi va operatorlarning ro'yxati chiqadi. Konkret funksiya bo'yicha yordamni olish uchun help <funksiya nomi, o'zgaruvchi, operator> komandasidan foydalaniladi.

Agar funksiya, operator yoki o'zgaruvchilarni nomi ma'lum bo'lmasa va biror kalit so'z ma'lum bo'lsa, kerakli faylni quyidagi komanda yordamida topamiz:

look for <kalit so'z>.

Bu komanda yordamida kalit so'z yordam tizimining barcha bo'limlarida qidiriladi va shu so'z bor bo'lim yoki fayl ko'rsatiladi.

1.3. Matlabning asosiy ob'ektlari, funksiyalari va sozlangan fuyrsiyalari

Har qanday matematik tizim kabi Matlabning ham asosiy markaziy tushunchasi matematik ifodalardir. Ma'lumki, har qanday matematik ifoda sonlar, konstantalar, o'zgaruvchilar, operatorlar, funksiyalar va turli xil maxsus matematik belgilar ustiga quriladi. Matlab da ham matematik ifoda xuddi shunday tarzda quriladi va asosiy ishlatiladigan obyektlardan biri hisoblanadi.

Matlabning asosiy obyektlari sifatida matematik ifodalar, sonlar, konstantalar, o'zgaruvchilar, operatorlar, funksiyalar va turli xil maxsus matematik belgilarni, hamda Matlab ning o'ziga xos boshqa tushunchalarini keltirish mumkin.

1) Konstantalar, o'zgaruvchilar va operatorlar

Son – Matlabning eng oddiy obyektidir. Ma'lumki, son miqdoriy ma'lumotlarni ifodalab beradi. Sonlar haqiqiy va kompleks bo'lishi mumkin.

Haqiqiy sonlar butun, kasr, fiksirlangan va suzuvchi nuqtali bo'lishi mumkin. Ularni Matlabda mantissa va son tartibini ko'rsatgan holda quyidagicha ifodalash mumkin:

0.4 -3.2 342 5.2e-24 -23.43e10

Ko'rinib turibdiki, mantissada sonning butun qismi kasr qismidan "nuqta" (.) orqali ajratiladi. Son tartibini mantissadan ajratish uchun "e" belgisi qo'yiladi, "+" ishora son oldiga qo'yilmaydi, "-" ishora esa qo'yiladi va u "unar" minus deb ataladi. Sonlarni ifodalashda raqamlar orasiga bo'sh joy ("probel") qo'yish mumkin emas.

Matlabda sonlarni ifodalash uchun quyidagi formatlardan foydalaniladi:

format bank, format short, format short e, format long, format long e, format rat.

Masalan, $x=[4/3, 1.234e-6]$ vektor uchun formatlarni e'lon qilib ko'ramiz:

```

>> x=[4/3 1.234e-6];
>> format bank
>> x
x=1.33 0.00
>> format short
>> x
x=1.3333 0.0000
>> format short e
>> x
x=1.3333e+000 1.234e-006
>> format long e
>> x
x=1.3....38E+00 1.2340....0E-006
>> format rat
>> x
x=4/3 1/810373

```

Bu formatlarning berilishi faqat natijaviy ma'lumotlarning ko'rinishiga ta'sir etadi. Barcha hisoblashlar ikki karrali (binar) aniqlikdagi formatda bajariladi, sonni kiritish esa ixtiyoriy qulay formatda bo'lishi mumkin.

Agar son kompleks bo'lsa, haqiqiy ($\text{Re}(z)$) va mavhum ($\text{Im}(z)$) qismlarga bo'linadi: $z = \text{Re}(z) + \text{Im}(z)i$. Mavhum qism kvadrat darajasi -1 ga teng bo'lgan i yoki j ko'paytuvchiga ega bo'ladi:

$2+3i$, $-3.141j$, $-123.456+2.7e-3i$ va h.k.

Matlabda z kompleks sonining haqiqiy qismlarini "real (z)", mavhum qismini "imag(z)", modulini "abs(z)", fazasini "angle(z)" funksiyalari ajratib beradi.

Masalan:

```

>> i
ans=0+1.0000i
>> z=2+3i
z=2.0000 + 3.0000i;
>> abs(z)
ans = 3.6055
>> real (z)
ans=2
>> a=imag(z)
a=3
>> b=angle(z)

```

b=0.9828

Matlabda konstanta (o'zgarimas) – bu avvaldan aniqlangan sonli yoki belgili qator bo'lib, u “noyob nom” (identifikator) bilan taqdim etiladi. Xususan, sonlar nomsiz sonli konstanta hisoblanadi.

Matlabda boshqacha ko'rinishdagi konstantalarni “tizim o'zgaruvchilari” deb atash qabul qilingan. Buning sababi, bir tomondan tizim yuklanayotgan vaqtda ular ham beriladi, ikkinchi tomondan dasturlarda bu “o'zgaruvchilar” qayta aniqlanishi mumkin.

Quyida asosiy tizim o'zgaruvchilarini keltirib o'tamiz:

- i yoki j – mavhum birlik;
- pi – $\pi=3.1415926$ soni;
- eps – sonlar ustida amallar bajarishdagi xatolik ($=2^{-52}$);
- realmin – suzuvchi nuqtali eng kichik son ($=2^{-1022}$);
- realmax – suzuvchi nuqtali eng katta son ($=2^{1023}$);
- inf – mashina cheksizlik qiymati;
- NaN – ma'lumotni sonli tavsifga ega emasligini ko'rsatuvchi o'zgaruvchi (Not a number);
- ans – qiymati boshqa o'zgaruvchiga o'zlashtirilmagan amalning natijasini saqlovchi o'zgaruvchi;
- belgili konstanta – bu apostrof ichiga olingan belgilar ketma-ketligi. Masalan, 'haqiqiy son', '3x+4y' va h.k.

Matlabda “umumiy o'zgaruvchilar” ham mavjud. Ular nomga ega obyektlar hisoblanadi. Bunday o'zgaruvchilarda turli xil qiymatlarni saqlash mumkin. O'zgaruvchilar sonli, belgili, vektorli yoki matritsali bo'lishi mumkin, lekin ularning hammasi Matlabda matritsa deb hisoblanadi.

Matlabda o'zgaruvchi turi e'lon qilinmaydi, balki u qiymatlariga qarab aniqlanaveradi. Demak, qiymat vektor yoki matritsa, sonli yoki belgili bo'lsa, o'zgaruvchi turi ham shunga mos bo'ladi.

O'zgaruvchi nomi (identifikator) boshlanishi harfdan iborat ixtiyoriy sondagi belgilardan iborat bo'lishi mumkin, ammo dastlabki 31 ta belgi bilan aniqlanadi (identifikatsiya qilinadi). Nom harfdan boshlansa-da, orasida harflar, raqamlar va “_” belgi (podcherkivanie) ishtirok etishi mumkin, lekin maxsus belgilar, masalan “+”, “-”, “*”, “/” va boshqalar qo'yish mumkin emas. Masalan, a1y23-o'zgaruvchi nomi bo'la oladi, lekin 2a1y23, a1/a2 – bo'la olmaydi.

Tabiiyki, o'zgaruvchi nomi boshqa o'zgaruvchilar nomlari bilan ustma-ust tushmasligi, yani “noyob nom” bo'lishi lozim.

Matlab dasturlash tilida o'zgaruvchiga qiymat berish quyidagi

<o'zgaruvchi nomi>=<ifoda>

komanda yordamida amalga oshiriladi, bu yerda "=" --qiymat berish, tayinlash operatori hisoblanadi.

Masalan,

```
>> x=5+exp(3);
```

O'zgaruvchi nomi oddiy yoki indekslangan bo'lishi mumkin. Matlabda o'zgaruvchilar nomi uchun lotin harflarini ishlatish tavsiya etiladi. Apostrof ichida kiritilgan simvollar ketma-ketligi simvolli o'zgaruvchilarni ifodalash uchun ishlatiladi.

Misol:

```
>>s='HUMO';
```

```
>>r='MATLAB';
```

```
>>v='6*3+4';
```

Matlabda operator deb ma'lumot(operand)lar ustida bajariladigan ma'lum bir amalning ijrosi uchun ishlatiladigan belgiga aytiladi.

Masalan, oddiy arifmetik amallarni ifodalovchi "+", "-", "*", "/" belgilar operatorlarga misol bo'ladi.

Matlabda barcha operatorlar ro'yxatini ko'rish uchun help ops komandasidan foydalaniladi.

2) Matlabda funksiyalar va sozlangan funksiyalar

Funksiya – o'zining argumentlari ustida ma'lum bir shakl almashtirishlarni bajaruvchi va hosil qilingan natijalarni qaytarish xususiyatiga ega noyob nomli obyektidir.

Agar funksiya bitta natijani qaytarsa, u matematik ifodalarda o'z nomi bilan ifodalanishi mumkin. Masalan, $\cos(x)$ funksiyani $4+3*\cos(3*\pi/4)$ ifodada to'q'ridan- to'q'ri ishlatish mumkin.

Ma'lumki, funksiya bir yoki ko'p argumentli bo'lishi mumkin. Bu holda argumentlar funksiya nomidan so'ng oddiy qavslar ichiga olib, vergullar bilan ajratilib ko'rsatiladi.

Agar funksiya bir nechta natijalarni qaytarsa, u quyidagicha ifodalanishi kerak:

$$[Y1, Y2, \dots, YN] = \text{func}(X1, X2, \dots, XM),$$

bu yerda N, M – chekli ma'lum sonlar, $X1, X2, \dots, XM$ – kirish parametrlari (argumentlari), $Y1, Y2, \dots, YN$ - chiqish parametrlari.

Matlabda funksiyalar shartli ravishda sozlangan (ichki) va tashqi yani m-funksiyalarga bo'linadi. Sozlangan (ichki) funksiyalarga elementar funksiyalar va maxsus funksiyalar kiradi. Elementar funksiyalarga $\sin(x)$, $\exp(x)$, tashqi funksiyalarga esa $\sinh(x)$ va boshqalarni misol qilib keltirish mumkin.

Elementar funksiyalar ro'yhati bilan komandalar oynasidan *help elfun*, maxsus funksiyalar ro'yhati bilan esa *help specfun* komandalari orqali tanishish mumkin.

Sozlangan funksiyalar Matlab tizimining kompilyatorlangan yadrosida, tashqi funksiyalar esa m-fayllarda saqlanadi.

Matlab sistemasi 1000 dan ortiq sozlangan funksiyalarga, o'nlab kengaytma paketlarda aniqlangan funksiyalarga ega. Shunday bo'lsa ham foydalanuvchi o'ziga kerakli yangi funksiyani hosil qilish va saqlab qo'yish imkoniyatiga ega.

Bunday imkoniyatni *inline-* funksiya va *handle* -anonim funksiyalar orqali yoki m-fayllarda amalga oshirish mumkin.

Nazorat savollari

1. Matlabning asosiy obyektlari nima?
2. Matlabda sonlarning qanday formatlari bor?
3. Matrisa, vektor-ustun va vektor-qatorni ta'riflang.
4. Matlabning eng sodda obyekti nima?
5. Kompleks songa boq'liq qanday funksiyalar mavjud?
6. Konstanta deganda nima tushuniladi?
7. Tizim o'zgaruvchilarining o'ziga xosligi nima?
8. O'zgaruvchilarni identifikasiyalash qanday amalga oshiriladi?
9. Simvulli o'zgaruvchilarga misollar keltiring.
10. Funksiyalarning sinflari haqida ma'lumot bering.

2. MA'LUMOTLARNI KIRITISH VA ODDIY HISOBLASH QOIDALARI

MATLAB tizimi shunday ishlab chiqilganki, hisoblashlarni foydalanuvchi dasturni tayyorlamasdan to'g'ridan-to'g'ri bajarishi mumkin. Bunda MATLAB superkalkulyator vazifasini bajarib, qatorli komanda rejimida ishlaydi.

2.1. Ma'lumotlarni(matritsalarini) kiritish

Ma'lumotlarni kiritish quyidagicha amalga oshiriladi:

- 1) boshlanq'ich ma'lumotlarni kiritishni ko'rsatish uchun “ >> ” belgidan foydalaniladi;
- 2) ma'lumotlar oddiy yozuvli tahrir yordamida kiritiladi;
- 3) hisoblash natijasini o'zlashtiruvchi o'zgaruvchi ko'rsatilmagan bo'lsa, MATLAB tizimi “ans” nomli o'zgaruvchini oladi;
- 4) kirish va hisoblash natijasini blokirovka qilish uchun “ ; ” (nuqtali vergul) qo'yiladi; agar natijani ko'rish lozim bo'lsa, o'zgaruvchi nomi yoki nom ko'rsatilmagan holda “ans” o'zgaruvchi chaqiriladi ;
- 5) o'zlashtirish amali sifatida ko'plab dasturlash tillari kabi “ := ” belgidan emas, balki oddiy tenglik “ = ” belgisidan foydalaniladi;
- 6) sozlangan funksiyalar (masalan, $\sin(x)$) yozma harflar bilan yoziladi, hamda ularning argumentlari oddiy qavs ichida yoziladi;
- 7) hisoblashlarning natijasi yangi qatorda “ >> ” belgisiz chiqadi;
- 8) muloqot “Savol berildi – javob olindi” yani dialogli ko'rinishda amalga oshiriladi;
- 9) komandali rejimda bir qatordagi belgilarning maksimal soni 4096, m-fayllarda esa chegaralanmagan;
- 10) agar ma'lumotlar bir qatorga sig'masa, u holda uchta yoki undan ko'p nuqtalar qo'yib, yangi qatorga o'tish mumkin;
- 11) ma'lumotlarni tashkillashtirish faqat matritsa ko'rinishida amalga oshiriladi(skalyar 1×1 o'lchamli, vektor-qator va vektor-ustun mos ravishda $1 \times n$ va $n \times 1$ o'lchamli matritsalar hisoblanadi. Maksimal o'lchov $n \times n$, $n = 2^{48} - 1$, bo'lishi mumkin);
- 12) agar o'zgaruvchi vektor (matritsa) bo'lsa, ko'pgina tizimlarda $\cos(v)$, $\exp(v)$ kabi funksiyalar ma'noga ega bo'lmaydi va hisoblanmaydi. Matlabda esa bu kabi hisoblar bajariladi va natija ham vektor (matritsa) bo'ladi.

Quyida oddiy hisoblashlar matematika va Matlabda ifodalanishiga doir misollar keltiramiz.

Matlabda	Matematikada
$2+3$	$2+3$
$2^{(3*\text{abs}(y/2))};$	$2^{3 y/2 }$
$2.301*\sin(x);$	$2,301\sin x$
$4+\exp(3)/5;$	$4+e^3/5$

2.2. Ma'lumotlar(matritsalar) va ularni shakllantirish usullari

Ma'lumki, Matlabda ma'lumotlar faqat bir shaklda, yani matritsa shaklida tashkil etiladi. Matritsalarini shakllantirishning 3 ta usuli mavjud:

1. Klaviatura orqali to'q'ridan – to'q'ri kiritish;
2. Faylli disklardan yuklash;
3. MATLAB komandalari yordamida hosil qilish.
 1. Klaviatura orqali kiritish.

Matritsalar kirish satridan kvadrat qavs “ [] ” orqali, elementlari orasiga vergul “ , ” yoki probel, satrlarni ajratish uchun nuqtali vergul “ ; ” qo'yib kiritiladi.

Misollar:

1) $a = (-1, 0, 4)$ vektor – satrni kiritish quyidagicha amalga oshirilsa bo'ladi:

- a) `>> a = [-1, 0, 4]`
- b) `>> a = [-1 0 4]`
- c) `>> a (1)=-1, a (2)=0, a (3)=4;`
`>> a`

Har bir holatda ish “ENTER” tugmasini bosish bilan tugallanadi. Hususan , `>> a` dan keyin “ENTER” tugmasi bosilsa , ekranda “ `>>` ” belgisiz $a = -1 0 4$ hosil bo'ladi.

2) (2×4) o'lchovli $y = \begin{pmatrix} 1 & 2 & -3 & 0 \\ 5 & 4 & 8 & -1 \end{pmatrix}$ matritsani kiritish uchun quyidagicha yo'l tutish lozim(probellar o'rniga vergul qo'ysa ham bo'ladi):

`>> y=[1 2 -3 0 ; 5 4 8 -1]`

Endi “ENTER” tugmasi bosilsa, ekranda:

$$y = \begin{matrix} 1 & 2 & -3 & 0 \\ 5 & 4 & 8 & -1 \end{matrix}$$

tasvir paydo bo'ladi.

Kerakli elementni chaqirish uchun unga indeksleri orqali murojaat qilish zarur. Masalan:

```
>> y(1,4)
```

So'ngra "ENTER" tugmasi bosilsa, ekranda :

```
ans = 0
```

 tasvir hosil bo'ladi va h.k..

Bu holatni elementlarning umumiy tartiblangan raqami orqali ham amalga oshirsa bo'ladi, u xolda elementlarning ketma-ket tartibi ustunlar bo'yicha hisoblanadi. Masalan:

```
>> y(4);
```

```
ans = 0
```

```
>> y(1)=y(8);
```

```
y(1) = -1
```

```
>> b=y(7);
```

```
b = 8
```

va h.k..

3) Berilgan matritsaning elementlarini o'zgartirish mumkin.

Masalan:

```
>> y(1,4)=10;
```

```
>> y
```

Endi "ENTER" tugmasi bosilsa, ekranda quyidagi tasvir ko'rinadi:

$$y = \begin{matrix} 1 & 2 & -3 & 10 \\ 5 & 4 & 8 & -1 \end{matrix}$$

4) Berilgan matritsani kengaytirish ham mumkin.

Masalan:

```
a) >> a1=[a 3 7]
```

U holda

a1 = -1 0 4 3 7 yangi vektor hosil bo'ladi.

```
b) >> a1(7)=8
```

U holda

a1 = -1 0 4 3 7 **0** 8 hosil bo'ladi (bu erda a1(6)=0 deb to'ldirilganiga

e'tibor bering!). Bu xossa matritsalar uchun ham o'rinli. Masalan,

```
>> y(10)=17;
```

```
>> y
```

```
y = 1 2 -3 10 0
```

```
5 4 -8 -1 17
```

Lekin u(9)=16 deb berilsa, tizim xatolik haqida axborot beradi.

```
c) >> yy=[y;11,13,-14,15]
```

Bu holda (3 × 4) o'lchovli matritsa hosil bo'ladi :

$$yy = \begin{bmatrix} 1 & 2 & -3 & 10 \\ 5 & 4 & 8 & -1 \\ 11 & 13 & -14 & 15 \end{bmatrix}$$

.>> c=[11;13];

>> yy1=[y,c]

U holda (2×5) o'lchovli matritsa hosil bo'ladi:

$$yy1 = \begin{bmatrix} 1 & 2 & -3 & 10 & 11 \\ 5 & 4 & 8 & -1 & 13 \end{bmatrix}$$

va h.k.

Matritsalarini kengaytirish, birlashtirish o'ziga xos qonuniyatga asoslanadi. Bu esa sal keyinroq ko'rsatiladi.

5) Matritsa elementlari ifoda ham bo'lishi mumkin. Masalan :

>> Z=[sin(0) sqrt(4) 2^3 + 1 5/2 3^2]

U holda ushbu vektor aniqlanadi:

Z=0.0000 2.0000 9.0000 2.5000 9.0000

>> a=[1; 0;-5^3] +i[3; sin(4);5]

a = 1.0e+002 *

0.0100 + 0.0300i

0 - 0.0076i

-1.2500 + 0.0500i

2. Ma'lumotlarni faylli disklardan yuklab ham hosil qilsa bo'ladi.

Buning uchun

load < fayl nomi >

komandasidan foydalaniladi. Agar komanda parametri yozilmasa, ma'lumotlar Matlab.mat faylidan yuklanadi. (Yuklanayotgan ma'lumotlar ASSII formatida yoki Matlabning ichki ikkilik(binar) formatida saqlanib qo'yilgan ham bo'lishi mumkin.) Kerakli ma'lumotlarni fayllardan tanlab-tanlab, masalan x,y,z matritsalarini, ham yuklab olish imkoniyati bor. Buning uchun

load < fayl nomi > x y z

komanda formatidan foydalaniladi.

3. Ma'lumotlarni MATLAB komandalari yordamida hosil qilish.

Matlabda ma'lumotlarni komandalar yordamida hosil qilishning bir nechta usullari bor. Shulardan biri ikki nuqta " : " komandasini qo'llashdir. Bu komanda sonlar ketma - ketligini (vektor - qatorlarni, vektor - ustunlarni, berilgan matritsalaridan yangi matritsa va vektorlar hosil qilish jarayonlarini) amalga oshirishda qulay hisoblanadi.

1) Ushbu $a = x1:h : x2$ komanda boshlanq'ich $x1$ qiymatdan h qadam bilan oxirgi qiymati $x2$ bo'lgan vektor - satrni hosil qiladi.

Misol.

>> $a = 2:0.5:5$

U holda $a = 2 \ 2.5 \ 3 \ 3.5 \ 4 \ 4.5 \ 5$ vektor hosil bo'ladi.

Agar h ko'rsatilmasa, u avtomatik ravishda 1 ga teng deb hisoblanadi.

Agar $x_1 > x_2$ bo'lib, $h > 0$ bo'lsa, tizim xatolik haqida ogohlantiradi.

Masalan,

>> $b = 5:0.5:2$

??? Error.....

2) Berilgan matritsadan vektor hosil qilish uchun quyidagi komanda formatlaridan foydalaniladi:

$y = x(:, <ustun\ nomeri>)$,

$yy = x(<satr\ nomeri>, :)$.

Misol .

>> $x = [2 \ 5 \ 7 \ -1 ; 4 \ -2 \ 1 \ 2 ; 0 \ 3 \ 4 \ -5]$

Undan so'ng "ENTER" tugmasi bosilsa, ekranda (3×4) o'lchovli ushbu matritsa ko'rinadi:

$$x = \begin{bmatrix} 2 & 5 & 7 & -1 \\ 4 & -2 & 1 & 2 \\ 0 & 3 & 4 & -5 \end{bmatrix}$$

>> $y = x(:,1)$

$$y = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}$$

3) Ikki nuqta (:) komandasini quyidagi formatlarda ham ishlatish mumkin:

$xy = x(:, k1:k2)$ - x matritsadan $k1$ dan $k2$ gacha bo'lgan ustunlar;

$yx = x(k3:k4, :)$ - x matritsadan $k3$ dan $k4$ gacha bo'lgan satrlar;

$xyx = x(k3:k4, :k1:k2)$ - x matritsadan $k3$ dan $k4$ gacha satrlar va $k1$ dan $k2$ gacha bo'lgan ustunlar kesishmasidagi elementlar ajratib olinadi va yangi matritsa sifatida e'lon qilinadi.

Misol. Yuqorida keltirilgan x matritsadan foydalanib, quyidagi hisoblashlarni amalga oshirish mumkin :

>> $xy1 = x(:,2:4)$

$$xy1 = \begin{bmatrix} 5 & 7 & -1 \\ -2 & 1 & 2 \\ 3 & 4 & -5 \end{bmatrix}$$

>>yx1=x(1:2,:)

$$yx1 = \begin{bmatrix} 2 & 5 & 7 & -1 \\ 4 & -2 & 1 & 2 \end{bmatrix}$$

>>xyx1=x(2:3,2:4)

$$xyx1 = \begin{bmatrix} -2 & 1 & 2 \\ 3 & 4 & -5 \end{bmatrix}$$

4) X – X matritsaning o'zini;

X(:) – X matritsaning elementlarini ustun ko'rinishida;

X(j:k) – X(j), X(j+1),...,X(k) ni ifodalaydi.

Ikki nuqta (:) komandasidan matritsa nafaqat 2 o'lchovli , balki katta o'lchovli holda ham foydalanish mumkin.

5) Matritsaning biror ustun yoki qatorini o'chirish orqali ham yangi matritsa hosil qilish mumkin. Buning uchun [] belgidan foydalanish kerak.

Misol.

>> x(:,1)=[]

$$\text{ans} = \begin{bmatrix} 5 & 7 & -1 \\ -2 & 1 & 2 \\ 3 & 4 & -5 \end{bmatrix}$$

>> x(2,:)=[]

$$\text{ans} = \begin{bmatrix} 2 & 5 & 7 & -1 \\ 0 & 3 & 4 & -5 \end{bmatrix}$$

Nazorat savollari

1. Ma'lumotlarni qanday kiritish usullari bor?
2. Ma'lumotlar qanday ko'rinishda tashkillashtiriladi?
3. Matritsa elementlari qanday bo'ladi?
4. load komandasining formatlarini va vazifalarini tushuntiring.
5. Ikki nuqta komandasining vazifalari nimalardan iborat?
6. Matritsalar qanday kiritiladi?
7. Matritsadan yangi matritsa qanday hosil qilinadi?
8. Ma'lumotlar faylli disklardan qanday yuklanadi?
9. Matritsa elementlariga murojaat qanday amalga oshiriladi?
10. Matritsalarining ustun va qatorlari qanday ajratiladi?

3. MATLABDA VEKTORLAR VA MATRITSALAR USTIDA AMALLAR

Matlabda matritsalar ustida arifmetik, mantiqiy va maxsus (kengaytma) amallar kirgizilgan. Albatta, ma'lumotlar turiga qarab bu amallar bajarilishi ma'lum bir talablar asosida quriladi.

3.1. Skalyar miqdorlar ustida arifmetik amallar

Amal nomi	Operator	Funksiya
Qo'shish	+	Plus
Ayirish	-	Minus
Ko'paytirish	*	Times
O'ngga bo'lish	/	Mrdivide
Chapga bo'lish	\	Mldivide
Darajaga oshirish	^	Mpower

Agar ifodada bir nechta amallar bo'lsa, ularni bajarilish ketma-ketligi quyidagi ustivorlik qoidasi bo'yicha amalga oshiriladi:

Ustivorlik	Amallar
1	(...) – oddiy qavs
2	^ - darajaga oshirish, chapdan o'ngga
3	Ko'paytirish va bo'lish, chapdan o'ngga
4	Qo'shish va ayirish, chapdan o'ngga

3.2. Matritsalar ustida oddiy arifmetik amallar

Matritsalar ustida oddiy arifmetik amallar bajarilishi uchun quyidagi talablar mavjud:

a) Qo'shish va ayirish amallari A va B matritsalarining mos elementlari orasida bajariladi. Shuning uchun A va B matritsalarining o'lchovi bir xil bo'lishi kerak:

$$A=[a(i,j)], \quad B=[b(i,j)], \quad S=[c(i,j)] \text{ bo'lsa,}$$

$$u \text{ holda } c(i,j)=a(i,j)\pm b(i,j), \quad i=\overline{1, n}, \quad j=\overline{1, m}.$$

Misol.

```
>>A=[1 2 3; 4 5 6];
```

```
>>B=[4 5 3; 2 3 -4];
```

```
>>S=A+B
```

```
S=
 5  7  6
 6  8  2
```

$$\gg d = A - B$$

$$d = \begin{bmatrix} -3 & -3 & 0 \\ 2 & 2 & 10 \end{bmatrix}$$

b) Matritsalarini ko'paytirish uchun chapdagi matritsaning ustunlari soni o'ngdagi matritsaning satrlari soniga teng bo'lishi kerak:

$A - (n \times k) -$ o'lchovli matritsa, $B - (k \times m) -$ o'lchovli matritsa bo'lsa, u holda

$S = A * B - (n \times m)$ o'lchovli matritsa bo'ladi va uning elementlari

$$c(i,j) = \sum_{l=1}^k a_{il} * b_{lj}, \quad i = \overline{1, n}, \quad j = \overline{1, m}$$

formula bo'yicha hisoblanadi. Masalan, $a = [1 \ 2; 0 \ 3; 2 \ 2]$, $b = [0 \ 1 \ 2; 3; 1 \ 0 \ 2 \ 3]$ bo'lsin. U holda $c = a * b$ quyidagicha bo'ladi:

$$c = [2 \ 1 \ 6 \ 9; 3 \ 0 \ 6 \ 9; 2 \ 2 \ 8 \ 1 \ 2].$$

c) Agar skalyar miqdor matritsaga ko'paytirilayotgan bo'lsa, u matritsaning har bir elementiga ko'paytiriladi:

$$k * A = [k * a(i, j)], \quad i = \overline{1, n}, \quad j = \overline{1, m}.$$

Masalan, $d = 3 * b$ bo'lsa, $d = [0 \ 3 \ 6 \ 9; 3 \ 0 \ 6 \ 9]$ hosil bo'ladi.

3.3. Matlabda massivlar ustida maxsus amallar

Matlab tizimida matritsalarining mos elementlari orasida yani massivlar ustida bajariladigan maxsus amallar kiritilgan. Bu amallarni ajratib ko'rsatish uchun belgi oldiga "nuqta" (.) qo'yiladi:

- 1) $A.^k - A$ matritsaning har bir elementi k darajaga ko'tariladi;
- 2) $A.*B - A$ ning har bir elementi B ning mos elementiga ko'paytiriladi;
- 3) $A./B - A$ ning har bir elementi B ning mos elementiga bo'linadi;
- 4) $A.\B - B$ ning har bir elementi A ning mos elementiga bo'linadi;
- 5) $A.^B - A$ ning har bir elementi B ning mos elementiga teng darajaga ko'tariladi.

Ko'rinib turibdiki, bu amallar bajarilishi uchun ham A va B matritsalar o'lchamlari teng bo'lishi kerak. Masalan, $a = [1 \ 2 \ 3; 2 \ 3 \ 1]$ va $b = [0 \ 1 \ 2; 2 \ 1 \ 2]$ bo'lsin. U holda $c = a.*b$ quyidagi matritsa bo'ladi:
 $c = [0 \ 2 \ 6; 4 \ 3 \ 2]$

3.4. Vektorlar ustida maxsus amallar

Matlab da vektorlar uchun "ichki ko'paytma" va "tashqi ko'paytma" deb ataluvchi amallar kirgizilgan. Bizga $a = [a_1, a_2, \dots, a_n]$ va $b = [b_1, b_2, \dots, b_n]$ vektorlar berilgan bo'lsin.

1-Ta'rif. a va b vektorlarning ichki (skalyar) ko'paytmasi deb $a*b = \sum_{i=1}^n a_i b_i$ kattalikka aytiladi.

2-Ta'rif. a va b vektorlarning tashqi ko'paytmasi deb elementlari $s_{ij} = a_i * b_j, i=\overline{1, n}, j=\overline{1, n}$, ko'paytmadan iborat bo'lgan $(n \times n)$ o'lchovli matritsaga aytiladi va u quyidagicha belgilanadi: $c = a * b$.

Masalan:

```

Command Window
>> a=2:10
a =
    2    3    4    5    6    7    8    9   10

>> b=0.5:0.2:2.2
b =
    0.5000    0.7000    0.9000    1.1000    1.3000    1.5000    1.7000    1.9000    2.1000

>> c=a*b'
c =
    82.2000

>> c1=a'*b
c1 =
    1.0000    1.4000    1.8000    2.2000    2.6000    3.0000    3.4000    3.8000    4.2000
    1.5000    2.1000    2.7000    3.3000    3.9000    4.5000    5.1000    5.7000    6.3000
    2.0000    2.8000    3.6000    4.4000    5.2000    6.0000    6.8000    7.6000    8.4000
    2.5000    3.5000    4.5000    5.5000    6.5000    7.5000    8.5000    9.5000   10.5000
    3.0000    4.2000    5.4000    6.6000    7.8000    9.0000   10.2000   11.4000   12.6000
    3.5000    4.9000    6.3000    7.7000    9.1000   10.5000   11.9000   13.3000   14.7000
    4.0000    5.6000    7.2000    8.8000   10.4000   12.0000   13.6000   15.2000   16.8000
    4.5000    6.3000    8.1000    9.9000   11.7000   13.5000   15.3000   17.1000   18.9000
    5.0000    7.0000    8.9000   11.0000   13.0000   15.0000   17.0000   19.0000   21.0000

```

3.1 - rasm. Vektorlarni ichki va tashqi ko'paytmasi.

3.5. Mantiqiy amallar

Mantiqiy amallarni ikki guruhga bo'lib o'rganamiz: solishtirish amallari va haqiqiy mantiqiy amallar.

Solishtirish amallariga quyidagilar kiradi:

- $a > b$ – “katta” amali;
- $a < b$ - “kichik” amali;
- $a <= b$ – “kichik yoki teng” amali;
- $a >= b$ – “katta yoki teng” amali;
- $a == b$ – “teng” amali;
- $a \sim b$ – “teng emas” amali;

Massivlarni solishtirishda bu amallar ularning mos elementlari orasida bajariladi. Natijada massivlar o'lchoviga teng o'lchovli massiv hosil bo'ladi. Agar solishtirish natijasi “rost” bo'lsa, massivning mos elementi 1 bo'ladi; agar solishtirish natijasi “yolq'on” bo'lsa, 0 bo'ladi.

Massivlarni solishtirishda $>$, $<$, $<=$, $>=$ amallari ishlatilsa, elementlarning faqat haqiqiy qismi solishtiriladi; $==$ yoki \sim amallari

ishlatilsa, u holda elementlarning ham haqiqiy, ham mavhum qismlari solishtiriladi.

Ikkita qator ekvivalentligini tekshirish uchun `strcmp` komandasidan foydalaniladi.

Solishtirish amali skalyar va matritsa ustida bajarilayotgan bo'lsa, skalyar matritsaning o'lchoviga teng matritsaga to'ldiriladi va undan keyin solishtiriladi.

Misollar:

```
1) >> a=3;
    >>b=[1 4 0; 2 5 7];
    >>a>b
    ans = 1 0 1
         1 0 0
```

2) Matritsa elementlari kompleks bo'lsin:

```
>> c = [5 + 2i 4 - i];
>>d = [5 + 7i 3 - i];
>>d <=c
    ans = 1 1
>>c <=d
    ans = 1 0
```

3) Solishtirish amallarini simvolli ifodalarga ham qo'llash mumkin.

```
>> `a` > `b`
    ans = 0
>> `c b a` < `a b c`
    ans = 0 1 0
```

Haqiqiy mantiqiy amallarga quyidagilar kiradi:

& - "va" amali

| - "yoki" amali

~ - "inkor" amali

Mantiqiy amallar matritsalarining mos elementlari orasida bajariladi. Agar amal natijasi "yolg'on" bo'lsa, u holda 0 ishlatiladi, "rost"ni bildiruvchi mantiqiy 1 ixtiyoriy nol bo'lmagan son bo'lishi mumkin.

Mantiqiy amallar uchun quyidagicha "rostlik" jadvali mavjud:

X	Y	X&Y	X Y	~X
0	0	0	0	1
0	1	0	1	1
0	1	0	1	1
1	1	1	1	0

Haqiqiy mantiqiy amallar bajarilishi bo'yicha arifmetik va solishtirish amallariga nisbatan past ustivorlikka ega. Mantiqiy amallar o'z- o'ziga nisbatan quyidagi ustivorlik qoidasiga bo'ysunadi: "inkor" eng yuqori ustivorlikka ega amal hisoblanadi; "va" bilan "yoki" amallari teng ustivorlikka ega va chapdan o'ngga ketma-ket bajariladi.

Misollar. 1) $1 \& 0 + 2$ bo'lsin. Bu erda dastlab $0 + 2 = 2$ bajariladi, keyin $1 \& 2 = 1$ natija olinadi.

2) $3 > 5 \& 1 | 0$ berilgan bo'lsin. Bu erda avval $3 > 5$ amal bajarilib, 0 natija olinadi. Keyin $0 \& 1 | 0$ ifodada chapdan o'ngga: $0 \& 1 = 0$, $0 | 0 = 0$ bajarilib, 0 natija olinadi.

Mantiqiy amallarga qo'shimcha ravishda yana quyidagilarni keltirish mumkin:

xor – "yoki" ni bekor qiluvchi amal;

any – "rost", agar vektorning barcha sonlari nolga teng bo'lsa;

all – "rost", agar vektorning barcha sonlari nolga teng bo'lmasa;

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

```
>> b = [1 0 0]
```

```
>>xor [a, b]
```

```
ans = 0;
```

```
>> any (a)
```

```
ans = 0
```

```
>> all (a)
```

```
ans = 1
```

```
>> all (b)
```

```
ans = 0
```

```
>> 'a b c' & '0 1 2'
```

```
ans = 1    1    1
```

3.6. Matritsalarini almashtirish amallari

Matlabda matritsalar ustida oddiy arifmetik va mantiqiy amallardan tashqari maxsus amallar va almashtirishlar mavjud. Bularga transponirlash, birlashtirish (konkatenatsiya) va burish amallari kiradi.

1) Transponirlash amali. Berilgan A matritsani transponirlash deganda uni mos qatorlarini ustunlar bilan almashtirish tushuniladi va u A' kabi belgilanadi (yoki transpose(A) komandasi orqali amalga oshiriladi).

Masalan, $A=[1, 2, 3; 4, 5, 6]$ matritsani transponirlasak, $A'=[1, 4; 2, 5; 3, 6]$ ko'rinishdagi (3x2) o'lchovli matritsa hosil bo'ladi.

2) Birlashtirish (konkatenatsiya) amali (cat). Bir nechta matritsalarini birlashtirish uchun “cat” komandasining quyidagi formatlaridan foydalaniladi:

cat (1, A, B) yoki **[A; B]** – vertikal birlashtirish (A va B matritsalarining ustunlari soni teng bo’lish kerak);

cat (2, A, B) yoki **[A, B]** – gorizontal birlashtirish (A va B matritsalarining satrlari soni teng bo’lishi kerak).

Bu qoidalar bajarilmagan hollarda tizim xatolik haqida axborot beradi.

Misol. `>>A=[1 0 3; 4 -3 6]; C=[7 -5 -3;8 11 10];`

`>>B=[2 1; 5 -7];`

$$\gg \text{cat}(1, A, C) = \begin{pmatrix} 1 & 0 & 3 \\ 4 & -3 & 6 \\ 7 & -5 & -3 \\ 8 & 11 & 10 \end{pmatrix}$$

$$\gg \text{cat}(2, B, A) = \begin{pmatrix} 2 & 1 & 1 & 0 & 3 \\ 5 & -7 & 4 & -3 & 6 \end{pmatrix}$$

`>> cat (1, C, B)`

“Error using vertcat. Cat arguments dimensions are not consistent”, ya’ni vertikal birlashtirish komandasi noto’g’ri ishlatilayotganligi, C va B matritsalarining o’lchovlari to’g’ri kelmasligi haqida axborot berilmoqda.

3) Burish amallarini bajarish uchun Matlabda quyidagi komandalardan foydalaniladi:

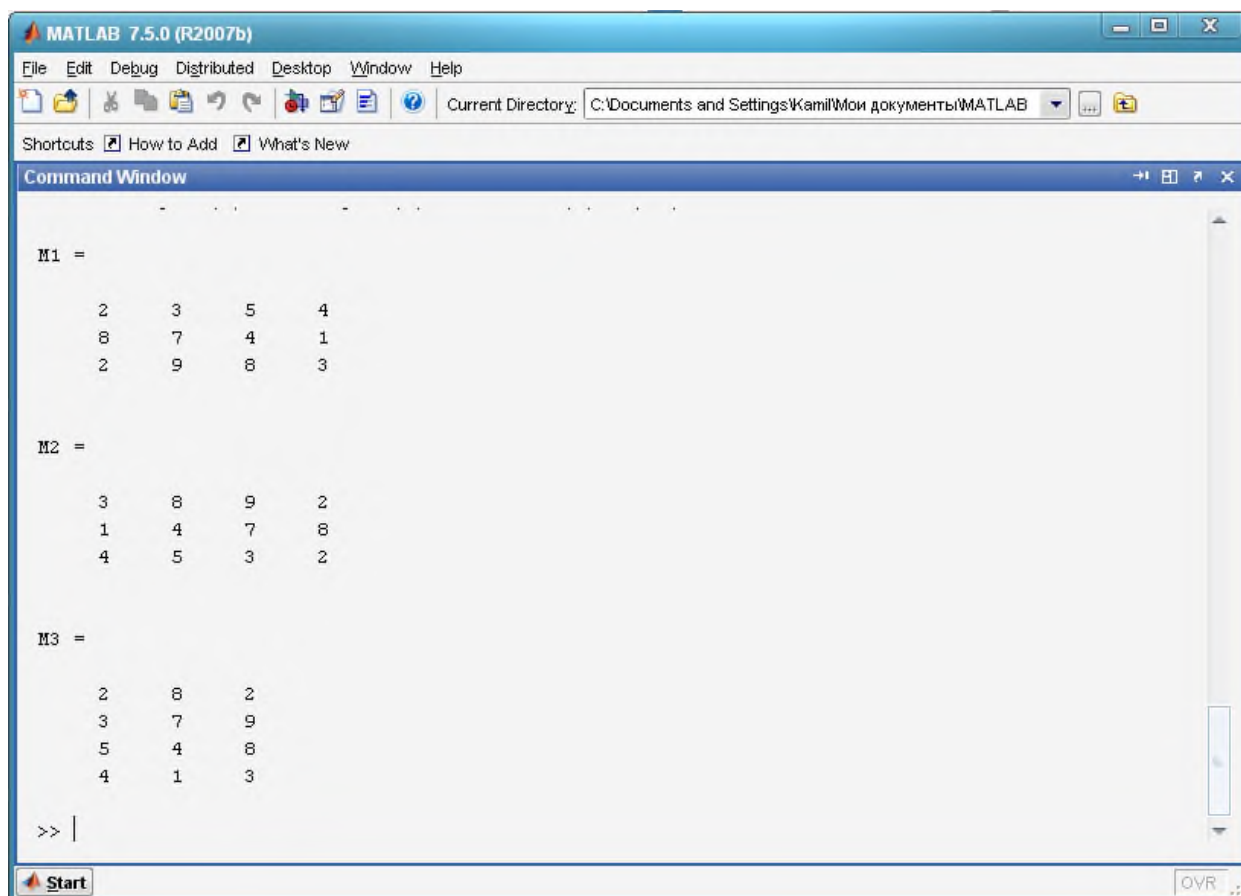
fliplr (A) – vertikal o’qqa nisbatan ustunlar almashtiriladi;

flipud (A) – gorizontal o’qqa nisbatan qatorlar almashtiriladi;

rot90 (A) – soat strelkasiga qarshi 90° ga buradi;

rot90 (A, k) – A matritsani 90*k° gradusga buradi (k – butun son).

Misol. `M=[4 5 3 2; 1 4 7 8; 3 8 9 2]; M1= fliplr (M);M2= flipud (M); M3= rot90 (M)`. Bu komandalardan keyin ENTER tugmasi bosilsa, quyidagi natija chiqadi:



3.2-rasm. Matritsalarini burish komandalarini natijasi.

Matlabda yana matritsalar ustida amallarni bajarish uchun quyidagi funksiyalar mavjud:

1) **sum(X)**- X matritsaning ustun bo'yicha elementlari yig'indilaridan tuzilgan vektor-satrni qaytaradi (agar X vektor bo'lsa, elementlar yiq'indisini hisoblaydi);

2) **sum(X,dim)**- agar dim=1 bo'lsa, xuddi sum(X) kabi; agar dim=2 bo'lsa, satrlar bo'yicha elementlar yiq'indisini hisoblaydi;

3) **det(X)**- X matritsaning determinantini qaytaradi;

4) **rank(X)**- X matritsaning rangini hisoblaydi;

5) **inv(X)**- X ga teskari matritsani hisoblaydi;

6) **prod(X)**- X matritsaning ustun bo'yicha elementlari ko'paytmalarining vektor-satirini qaytaradi (agar X vektor bo'lsa, elementlari ko'paytmasini);

7) **prod(X,dim)** – agar dim=1 bo'lsa, prod(X) kabi; agar dim=2 bo'lsa, satrlar bo'yicha elementlar ko'paytmasi hisoblanadi;

8) **tril(X)**- X matritsaning asosiy diagonaldan pastda turgan qismini o'zgarishsiz, yuqori qismini nollarga almashtirib qaytaradi;

9) **triu(X)**- X matritsaning asosiy diagonaldan yuqorida turgan qismini o'zgarishsiz, pastki qismini nollarga almashtirib qaytaradi;

10) **sort(X)**- agar X vektor bo'lsa, elementlarini o'sish tartibida joylashtiradi; X-matritsa bo'lsa, ustun bo'yicha o'sish tartibida sortirovka qiladi;

11) **[B,index]=sort(X)**-sortirovka qilingan massiv bilan birga indekslar massivini ham qaytaradi (ustundagi o'rniga qarab);

12) **sort(X, dim)** - dim ning qiymatiga qarab cortirovka amalini bajaradi;

13) **max(X)**- X matritsani ustun bo'yicha eng katta elementlaridan iborat vektor-satrni qaytaradi;

14) **max(X,Y)** – X va Y massivlarning mos elementlari solishtiriladi va ularning kattalaridan iborat massiv qaytariladi;

15) **max(X,[], dim)** – dim ning qiymatiga bog'liq ravishda ishlaydi (dim=1,2);

16) **[S, I] =max(X)**- maksimum qiymatlardan tashqari ularning indekslarini ham beradi (ustundagi o'rni bo'yicha);

17) **min(X)** va uning boshqa formatlari xuddi **max(X)** ga o'hshash,faqat minimumga nisbatan;

18) **mean(X)**- X matritsaning ustun bo'yicha elementlari o'rta qiymatlari hisoblanadi (X vektor bo'lsa, elementlarining o'rta qiymatini qaytaradi);

19) **mean(X,dim)**- dim ni qiymatiga bog'liq ravishda ishlaydi (dim=1,2);

20) **trace(X)** –X matritsaning diagonal elementlari yig'indisi (X matritsaning izi) ni qaytaradi ;

21) **repmat(X,n,m)** – X matritsani vertikal n marta , gorizonta m marta takrorlagan holda matritsa hosil qiladi;

22) **diag(X)** – a) agar X matritsa bo'lsa, diagonal elementlaridan iborat vektor-satrni qaytaradi; b) agar X vektor bo'lsa, diagonali X ning elementlaridan , qolgan elementlari nollardan iborat kvadrat matritsa yasaydi .

Matlabda maxsus ko'rinishdagi matritsalarini hosil qiluvchi komandalar ham mavjud. Ularga misol sifatida quyidagilarni keltirish mumkin:

- **eye (m, n)** – asosiy diagonalda 1, qolgan elementlari 0 bo'lgan (mxn) o'lchovli matritsa hosil qilinadi;

- **linspace (a, b, [n])** – [a, b]-oralikda tekis taqsimlangan n ta elementli vektorni aniqlaydi (n ko'rsatilmasa,avtomatik tarzda 100 deb olinadi);

- **ones (m, n)** – elementlari faqat 1 dan iborat bo'lgan (mxn)- matritsa;

- **rand (m, n)** – elementlari (0, 1) oraliqda tekis taqsimlangan tasodifiy miqdorlar bo'lgan (mxn) -o'lchovli matritsa;

- **randperm(n)** – 1dan n gacha bo'lgan butun sonlarning tasodifiy taqsimlangan vektor-satrini qaytaradi;

- **zeros (m, n)** – (mxn) o'lchovli faqat nollardan tuzilgan matritsa;

- **hilb (n)** – n-tartibli Gilbert matritsasi (elementlari $h(i, j)=1/(i+ j)$);

- **invhilb (n)** – Gilbertning teskari matritsasi;

- **magic (n)** – qator bo'yicha elementlar yig'indisi ustun bo'yicha elementlar yig'indisiga teng bo'lgan “sexrli” matritsa;

- **size (A)** – A matritsaning o'lchovi;

- **length (A)** – A vektor uzunligi (elementlar soni); A matritsa uchun max (mxn) ni beradi;

- **ndims (A)** – A matritsa o'lchovlari soni;

- **isempty (A)** – A matritsa bo'sh bo'lsa, 1 ni, aks holda 0 ni qaytaradi;

- **isequal (A, B)** – $A=B$ bo'lsa, 1 ni, aks holda 0 ni qaytaradi;

- **isnumeric (A)** – A matritsa sonli tip bo'lsa, 1ni, aks holda 0 ni beradi;

- **pascal (n)** – Paskal matritsasini beradi.

Ta'kidlash joizki, Matlab tizimida yana bir qancha maxsus matritsalar mavjud.

3.7. Sana va vaqt funksiyalari

Matlabda sana va vaqt funksiyalarining bir nechta formatlari kiritilgan. Quyida shu formatlarning ba'zi ko'rinishlarini keltiramiz:

1. **calendar**–joriy oy kalendarining (6x7) matritsa ko'rinishini qaytaradi. Kalendar yakshanba (birinchi ustun) dan boshlanib shanba bilan tugaydi.

2. **calendar (d)** – d songa mos keluvchi kunni o'z ichiga oluvchi oy kalendarini qaytaradi (kunlar hisob boshi(letopis)dan boshlab sanaladi).

3. **calendar (y,m)** – y argument bilan ko'rsatilgan yil va m argument bilan berilgan oy kalendarini qaytaradi.

3. **clock** – 6 elementli vektorni qaytaradi (yil oy kun soat minut sek). Bu komanda bajarilgandan so'ng fix komandasini qo'llash kerak.

4. **str=date** – sanani dd-mmm-gggg formatda ko'rsatuvchi vektor - qatorni beradi.

5. **date num** – sananing qator ko’rinishini tartib raqamli ko’rinishga o’tkazadi (qandaydir boshlang’ich kundan boshlab: (01,01,00)) va x.k.

Misollar.

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\user\M
Shortcuts How to Add What's New

>> calendar

           May 2013
  S      M      Tu      W      Th      F      S
  0      0      0      1      2      3      4
  5      6      7      8      9     10     11
 12     13     14     15     16     17     18
 19     20     21     22     23     24     25
 26     27     28     29     30     31      0
 0      0      0      0      0      0      0

>> calendar(7800690)

           Jul 1357
  S      M      Tu      W      Th      F      S
  0      0      0      0      0      1      2
  3      4      5      6      7      8      9
 10     11     12     13     14     15     16
 17     18     19     20     21     22     23
 24     25     26     27     28     29     30
 31      0      0      0      0      0      0

>> a=clock

a =

 1.0e+003 *
 2.0130    0.0050    0.0210    0.0140    0.0050    0.0059

>> fix(a)

```

3.3 - rasm. Sana va vaqt funksiyalaridan foydalanish.

fix(a) komandasining natijasi quyidagicha:

ans= 2013 5 21 14 5 5

>>d=date

d=21-May- 2013

Nazorat savollari

1. Qanday arifmetik amallarni bilasiz?
2. Matritsalar ustida bajariladigan qanday amallar bor?
3. Arifmetik amallar bajarilishini ustuvorlik qoidasi qanday?
4. Matritsalar ustida ko’paytirish amali qanday bajariladi?

5. Matlabda qanday solishtirish va mantiqiy amallari mavjud?
6. Matritsani transponirlash nima?
7. flipud va fliplr komandalari nima uchun xizmat qiladi?
8. Matritsalarini birlashtirish qanday komanda orqali bajariladi?
9. Maxsus matritsa deganda nimani tushunasiz?
10. Maxsus matritsalarini hosil qiluvchi komandalarni keltiring.

4. MATLABDA SIYRAKLASHGAN MATRITSALAR

4.1. Siyraklashgan matritsalar ustida amallar bajarish

Elementlari nolga teng bo'lmagan matrisa to'la deyiladi. Siyraklashgan matrisa deb nol elementlarga ham ega bo'lgan matrisa tushuniladi. Siyraklashgan matritsalar bu ma'lumotlarni o'ziga xos saqlash sxemasi va zarur amalni bajarish uchun mos algoritmi bilan birgalikdagi majmuadir. Agar keltirilgan ma'lumotlarni saqlovchi sxema va algoritm massiv ko'rinishdagi oddiy saqlash sxemasi va oddiy algoritmga qaraganda xotira va vaqtdan yutish imkonini bersa, bu holda siyraklashgan matritsalardan foydalansak bo'ladi.

Siyraklashgan matritsalar turli xil masalalarda paydo bo'ladi. Ularni birlashtiruvchi xususiyat bitta: bu masalalarda noma'lumlar soni ko'p, ular tenglamalar orqali boq'langan, har bir boq'liqlikda faqat bir nechta noma'lumlar ishtirok etadi.

Siyraklashgan matritsalar Matlabda kompakt formada, yani faqat nol bo'lmagan elementlar va ularning mos indeksleri tasvirlanadi va saqlanadi. Matlabda siyraklashgan matritsalar bilan ishlash uchun qator sozlangan funksiyalar mavjud.

1. SPARSE funksiyasi yordamida siyraklashgan matritsa yaratish mumkin, yani nol elementli matritsani siyraklashgan ko'rinishda tasvirlaydi. Bu funksiyaning quyidagi formatlari mavjud:

S=sparse(i,j,s,m,n,nzmax) – (mxn)o'lchovli siyraklashgan matritsani (nol bo'lmagan elementlar soni nzmax dan katta emas) i, j, s vektorlardan foydalanib hosil qiladi(bu erda i, j –indekslarni aniqlaydi, s- element qiymatini aniqlaydi);

S=sparse(i,j,s,m,n) – bu erda avtomatik tarzda nzmax=length(s);

S=sparse(i,j) – bu erda m=max(i), n=max(j) funksiyalar s ning nol satrlari o'chirilmasdan avval hisoblanadi;

S=sparse(m,n) –siyraklashgan matritsa uchun xotirani band etadi va sparse([],[],[],m,n,0) funksiyaga teng kuchlidir (barcha mxn elementlar nolga teng);

S=sparse(A) – A matritsaning barcha nol elementlarini o'chirib, siyraklashgan matritsa hosil qiladi.

Misol:

```
A=[10 0 0; 0 0 20; 30 40 0];
```

```
>> S=sparse(A)
```

```
S =
```

(1,1) 10
 (3,1) 30
 (3,2) 40
 (2,3) 20

Keltirilgan misolda matritsani bunday saqlash sxemasining yutuq'i yaqqol ko'rinib turibdi. Bu misolda xotira faqat nolga teng bo'lmagan double turdagi sonlarga va uint32 turdagi indekslarga ajratiladi. A matritsa 72 bayt xotirani egallaydi, S siyraklashgan matritsa esa 64 bayt xotirani egallaydi. Bunday yutuq katta matritsalar uchun sezilarli bo'ladi.

Keyingi misolda SPARSE funksiyasi orqali siyraklashgan matritsa yaratilishining yana bir holi yoritiladi (nolga teng bo'lmagan sonlarning indekslari va ularning qiymatlari, hamda matritsa o'lchovi mos ravishda kiritiladi):

```
>> S = sparse([1 3 3 2], [1 1 2 3], [10 30 40 20], 3, 3)
```

S =

(1,1) 10
 (3,1) 30
 (3,2) 40
 (2,3) 20

2. SPDIAGS funksiyasi orqali faqat matritsaning diagonallarida nolga teng bo'lmagan elementlar joylashgan siyraklashgan matritsa yaratsa bo'ladi.

SPDIAGS funksiyasi kiruvchi parametrlar soniga qarab, 4 ta formatda ishlatiladi:

[B, d] = spdiags(A) –(mxn)-o'lchovli A matritsadan barcha nolmas diagonallarni ajratadi; B -min(m, n)xp - o'lchovli matritsa bo'lib, p ustunlari- A matritsaning nolmas diagonallari; d – p vektor uzunligi (p vektorning butun musbat elementi asosiy diagonaldan yuqoridagi diagonal nomerini, manfiy elementi esa pastki diagonal nomerini bildiradi);

B = spdiags(A, d) - d da ko'rsatilgan diagonallarni ajratadi;

A = spdiags(B, d, A)- A matritsaning d da ko'rsatilgan diagonallarini B matritsaning ustunlari bilan almashtiradi;

A = spdiags(B, d, m, n)- d da ko'rsatilgan diagonallar bo'yicha B matritsaning ustunlarini joylashtirib, (mxn) o'lchovli siyraklashgan matritsa yaratadi.

3. FULL funksiyasi. **full(s)** – s siyraklashgan matritsani to'la ko'rinishga keltiradi.

Misol.

```
>> S = sparse([1 3 3 2], [1 1 2 3], [10 30 40 20], 3, 3)
```

```
S =
```

```
(1,1)    10  
(3,1)    30  
(3,2)    40  
(2,3)    20
```

```
>> A = full(S)
```

```
A =
```

```
10  0  0  
0  0  20  
30  40  0
```

5. SPEYE funksiyasi. `speye(m, n)` - bosh diagonali birlardan, qolgan elementlari nollardan iborat ($m \times n$) o'lchovli siyraklashgan matritsa yaratadi;

speye(n) yoki **speye(n, n)** funksiyadir.

Misol.

```
>> S=speye(3)
```

```
S =
```

```
(1,1)    1  
(2,2)    1  
(3,3)    1
```

6. SPRAND funksiyasi. **R=sprand(m,n, density)**— $\text{density} \times m \times n$ ta tekis taqsimlangan nolmas elementlarga ega bo'lgan ($m \times n$) o'lchovli tasodifiy siyraklashtirilgan matritsani qaytaradi ($0 \leq \text{density} \leq 1$).

R = sprandn(S) - S siyraklashtirilgan matritsaning strukturasi o'xshash R matritsani yaratadi, biroq uning elementlari o'rta qiymati 0 ga va dispersiyasi 1 ga teng bo'lgan normal qonun bo'yicha taqsimlangan.

R = sprandn(m, n, density) - tasodifiy siyraklashgan matritsa, uning nolmas elementlar soni taxminan $\text{density} \times m \times n$ ga teng va ular normal qonuni bo'yicha taqsimlangan.

```
>> R=sprandn(S)
```

```
R =
```

```
(1,1)    0.2944  
(2,2)   -1.3362  
(3,3)    0.7143
```

7. SPRANDSYM funksiyasi. **R = sprandsym(S)** – tasodifiy simmetrik matritsa bo'lib, uning bosh diagonali va quyi diagonallarining strukturasi S matritsaning strukturasi o'xshash, elementlarining qiymatlari o'rta

qiymati 0 ga va dispersiyasi 1 ga teng bo'lgan normal qonuniyat asosida taqsimlangan.

Bu funksiyaning yana quyidagicha formatlari mavjud:

R = sprandsym(n, alpha) , R = sprandsym(n, alpha, rcond)

Misollar. >> M=[23 2 11; 0 23 1; 0 12 3]

M =

23 2 11

0 23 1

0 12 3

>> A=sprandsym(M)

A =

(1,1) 2.1832

(2,2) -0.1364

(3,2) 1.0668

(2,3) 1.0668

(3,3) 0.1139

8. FIND funksiyasi. **k = find(x)** - x vektorning nolmas elementlarining indekslarini aniqlaydi; agar bunday elementlar bo'lmasa, natija bo'sh vektor bo'ladi. Agar X kirish matritsa bo'lsa, bunday murojaatda u ustun-vektor deb qaraladi(bu ustun-vektor berilgan matritsaning ustunlar birlashmasidan tashkil topgan, deb hisoblanadi).

[i, j] = find(X) - X matritsa nolmas elementlarining satr va ustun indekslarini qaytarib beradi;

[i, j, s] = find(X) --indekslarni va X matritsaning nolmas elementlardan iborat s ustun-vektorini qaytaradi.

[i j]= find(X,N) – Xmatrisaning 1-ustunida joylashgan dastlabki N ta nolmas elementning indekslarini qaytaradi; **[i j]= find(X,N,'first')** – xuddi find(X,N) kabi;

[i j]= find(X,N,'last')- oxirgi ustundagi dastlabki N ta nolmas elementning indekslarini aniqlaydi; **[i j m]= find(X)**- X ning nolmas elementlari m vektor –ustun sifatida va ularning indekslarini qaytaradi;

[i j m]=find(X<munosabat operatori>N)-ko'rsatilgan munosabatni qanoatlantiruvchi elementlar indekslari va mantiqiy rostliklardan iborat vektor-ustunni hosil qiladi.

Misollar: To'rt o'lchovli sexrli matritsadan hamda buyruqlar oynasida hosil qilingan d matritsadan foydalanib, barcha yuqoridagi find operatorlarini ishlashini tekshirib ko'ring.

Matritsalarini hosil qilib quyidagi natijalarga ega bo'lamiz:

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\user\Mc
Shortcuts How to Add What's New

>> a=magic(4)

a =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

>> b=find(a)

b =
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

```

4.1 - rasm. $k = \text{find}(x)$ komandasining qo'llanishi.

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\user\Moi z
Shortcuts How to Add What's New

16

>> [i j]=find(a,3)

i =

     1
     2
     3

j =

     1
     1
     1

>> [i j]=find(a,3,'last')

i =

     2
     3
     4

j =

     4
     4

```

4.2 -rasm. Nolmas elementlar indeksleri.

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Shortcuts How to Add What's New
Current Directory: C:\D

>> [i j]=find(a,3,'last')

i =
     2
     3
     4

j =
     4
     4
     4

>> [i j]=find(a<3)

i =
     1
     4

j =
     2
     4

>>
Start

```

4.3 – rasm. Nolmas va shartli elementlar indekslari.

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Shortcuts How to Add What's New
Current Directory: C

>> d=[0.5 2 3 0;5 11 10 8;9 7 6 12 ;4 14 0 0];
>> [i j m]=find(d<3)

i =
     1
     1
     4
     1
     4

j =
     1
     2
     3
     4
     4

m =
     1
     1
     1
     1
     1

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window He
Shortcuts How to Add What's New
Current Directory: C

>> [i j m]=find(d); [i j m]

ans =

     1.0000     1.0000     0.5000
     2.0000     1.0000     5.0000
     3.0000     1.0000     9.0000
     4.0000     1.0000     4.0000
     1.0000     2.0000     2.0000
     2.0000     2.0000    11.0000
     3.0000     2.0000     7.0000
     4.0000     2.0000    14.0000
     1.0000     3.0000     3.0000
     2.0000     3.0000    10.0000
     3.0000     3.0000     6.0000
     2.0000     4.0000     8.0000
     3.0000     4.0000    12.0000

```

4.4 - rasm. Nolmas elementlar va ularning indekslari.

9. ISSPARSE funksiyasi. Agar X matritsa siyraklashgan bo'lsa, **issparse(x)** funksiya 1 ni qaytaradi, aks xolda 0 ni qaytaradi. Misol: `>> S = sparse([1 3 3 2], [1 1 2 3], [10 30 40 20], 3, 3)`

`S =`

`(1,1) 10`

`(3,1) 30`

`(3,2) 40`

`(2,3) 20`

`>> A=issparse(S)`

`A = 1`

10. NNZ funksiyasi yordamida X matritsaning nolmas elementlar sonini qaytarish mumkin.

Misol: `>> A=nnz(S)`

`A = 4`

11. NONZEROS funksiyasi yordamida X matritsaning ustunlar bo'ylab nolmas elementlarini tanlab vektor yaratish mumkin. Misol:

`>> A=nonzeros(S)`

`A = 10`

`30`

`40`

`20`

12. SPFUN funksiyasi - berilgan siyraklashgan matritsaning nolmas elementlariga ko'rsatilgan funktsiyani qo'llaydi.

Misol: `>> f = spfun(@sin, S)`

`f =`

`(1,1) -0.5440`

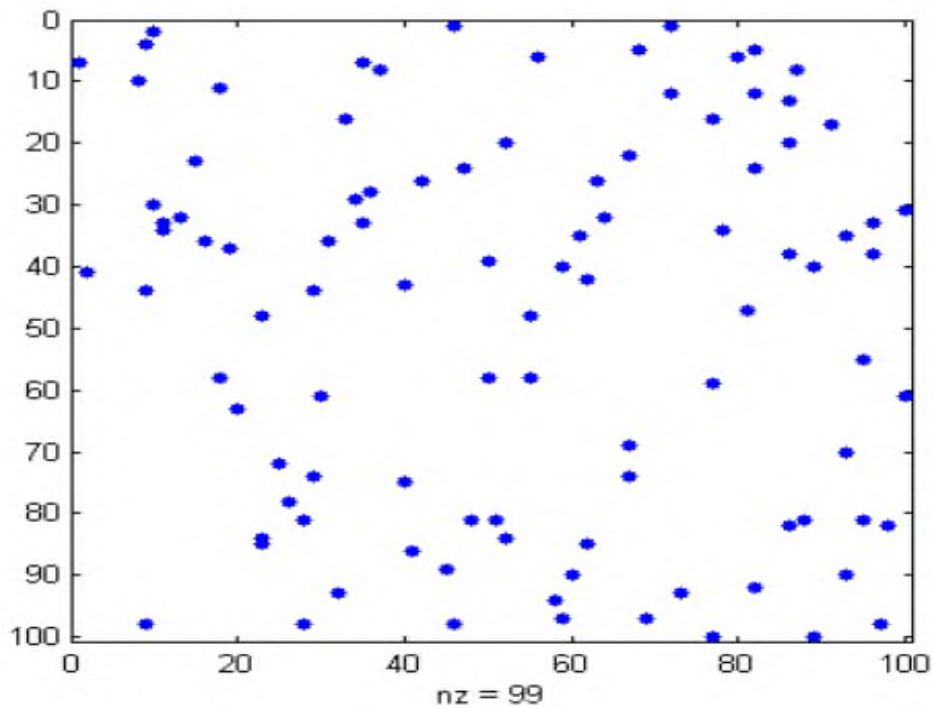
`(3,1) -0.9880`

`(3,2) 0.7451`

`(2,3) 0.9129`

13. SPY funksiyasidan siyraklashgan matritsalarini vizuallashtirish uchun foydalaniladi. (Siyraklashgan matritsa shabloni va nolmas elementlar sonini chiqaradi). Misol:

`S = sprand(100, 100, 0.01); spy(S)`



4.5 - rasm. Siyraklashgan matritsalarini vizuallashtirish.

Nazorat savollari

1. Qanday matritsa to'la deyiladi?
2. Siyraklashgan matritsa deb nimaga aytiladi?
3. Matlabda siyraklashgan matritsa deb nima tushuniladi?
4. Siyraklashgan matritsalar qachon hosil bo'lishi mumkin?
5. Matlabda siyraklashgan matritsalar ustida ishlash uchun qanday funksiyalar mavjud?

5. SIMVOLLI O'ZGARUVCHILAR ALGEBRASI

Ma'lumki, simvolli o'zgaruvchilar sinfi sonli o'zgaruvchilar sinfidan tubdan farq qiladi. Chunki sonli o'zgaruvchilar yordamida faqat arifmetik ifodalar qiymatlari hisoblansa, simvolli o'zgaruvchilar yordamida algebraik ifodalar ustida har xil almashtirishlar va amallar bajarish mumkin bo'ladi. Shuning uchun simvolli o'zgaruvchilar bilan ishlash Matlab tizimida bir nechta qulayliklarni hosil qiladi. Buni simvolli o'zgaruvchilar bilan ishlaydigan Matlabning quyidagi komandalari misolida ko'rish mumkin:

```
- a = sym('a'), b = sym('b'), c = sym('c')
```

- `syms a b c` - `a, b, c` - o'zgaruvchilarni simvolli deb elon qilish komandalaridir.

Masalan, komandalar oynasida x va y larni simvolli deb e'lon qilamiz. Undan keyin ularni qiymatini chiqarish komandasini bersak, sistema xatolikni beradi:

```
>>x=sym('x')
```

```
x = x
```

```
>>x
```

```
??? Undefined function or variable 'x'
```

5.1. Simvolli funksiyalar va ifodalar

Simvolli funktsiyani e'lon qilish uchun `y = sym('f(x)')` komandasini qo'llash kerak. Masalan, komandalar oynasida $y = ax^2 + bx + c$ funksiya ko'rinishini

quyidagicha hosil qilish mumkin:

```
>> y = sym('a * x^2 + b * x + c')
```

```
y = a * x^2 + b * x + c
```

Funksiyani berish uchun boshqa komandalardan ham foydalansa bo'ladi:

```
>> syms a b c x;
```

```
>> y = f(x, a, b, c)
```

Bu holda funksiya aniqlanishida ishlatilayotgan barcha simvolli o'zgaruvchilar avval e'lon qilinadi. Masalan, $y = ax^2 + bx + c$ simvolli funktsiyani aniqlash va unda $y_1 = y - c$, $f = cy$, $f_1 = y/c$, $g = y^a$, $g_1 = \sqrt{y}$ kabi almashtirishlarni bajarish kerak bo'lsa, quyidagi komandalardan foydalaniladi:

$$\text{syms } a \quad b \quad c \quad x$$

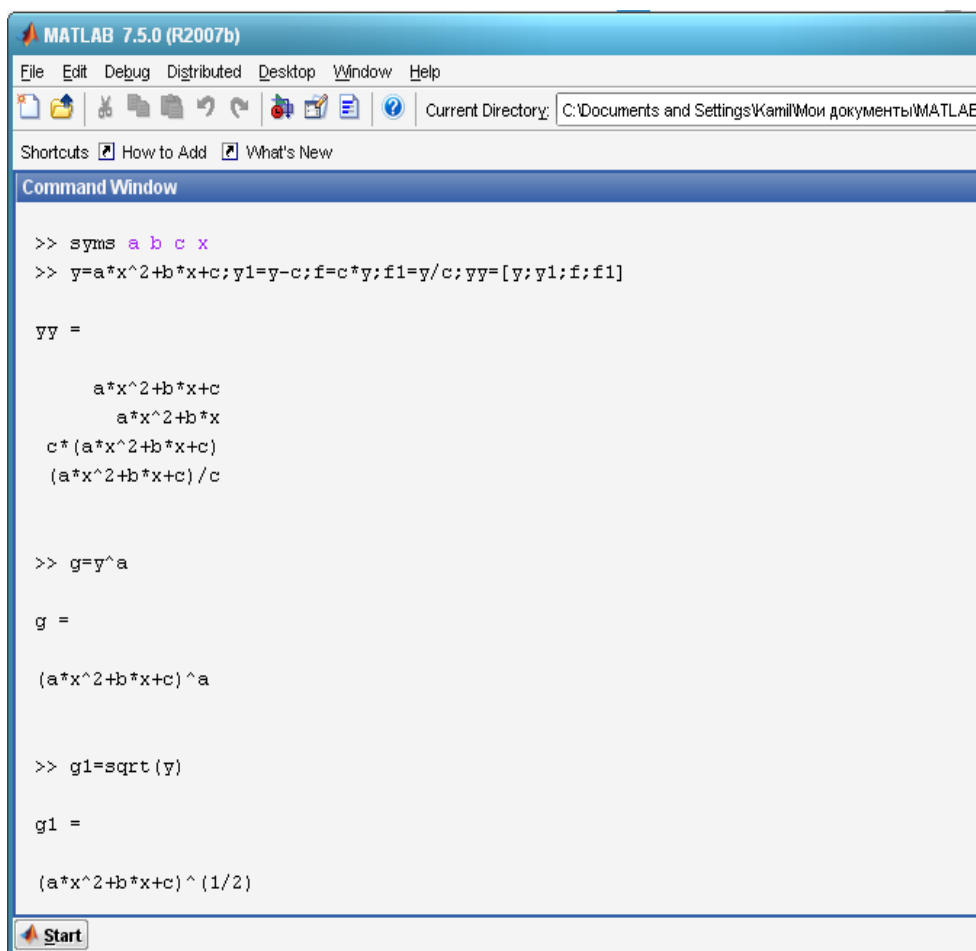
$$y = a * x^2 + b * x + c$$

$$y_1 = y - c, \quad f = cy, \quad f_1 = y / c,$$

$$g = y^a$$

$$g_1 = \text{sqrt}(y)$$

Natijalar ekranga chiqadi :



5.1 - rasm. Simvulli o'zgaruvchilar ustida amallar.

Ifodalar ustida quyidagi matematik operatsiyalarni bajarish mumkin:

- $p1 = \text{expand}(p)$ – p ifodani to'la yoyish komandasi;
- $p1 = \text{collect}(p, 'a')$ – p ifodani a ning darajalari bo'yicha yoyish komandasi;
- $p1 = \text{factor}(p)$ – p ifodani ko'paytuvchilarga ajratish komandasi;
- $p1 = \text{subs}(p, 'a', 'b')$ – p ifodada a o'zgaruvchining o'rniga b ni qo'yish komandasi (agar bir nechta a, c, d o'zgaruvchilarni almashtirish kerak bo'lsa, u holda $\{ 'a', 'c', 'd' \}$ kabi belgilash ishlatiladi);
- $p1 = \text{simplify}(p)$ – p ifodani soddalashtirish komandasi.

Misol. $p = (a + b)^4 + 3a^2b^4 - 4ab + c * a^3$ ko'phadni a va b ning darajalari bo'yicha va to'la yoying. Bu misolni quyidagi komandalar ketma-ketligi xal qilib beradi:

`syms a b c`

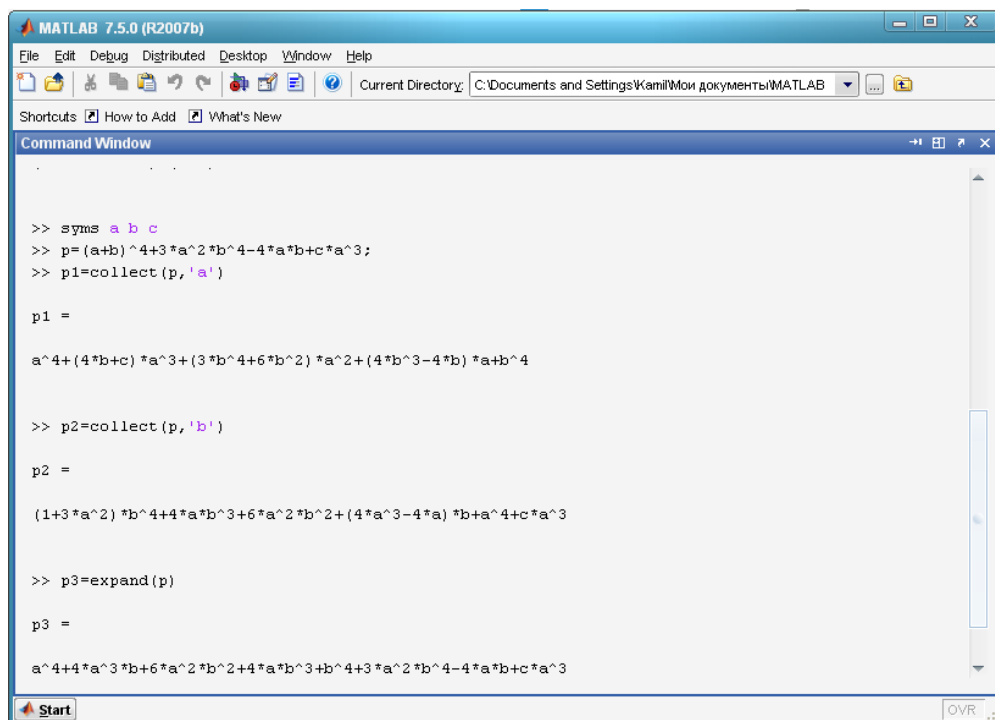
`p = (a + b)^4 + 3 * a^2 * b^4 - 4 * a * b + c * a^3;`

`p1 = collect(p, 'a')`

`p2 = collect(p, 'b')`

`p3 = expand(p)`

Natija:



5.2 - rasm. Simvolli funksiyalarning qo'llanilishi.

Misol. $p = (a + b)^3 + 4(a + b)^2 + 5(a + b) + 1$ ko'phadni ko'paytuvchilarga ajrating, $b = a + 1$ almashtirishni bajaring va uni soddalashtiring.

`syms a b`

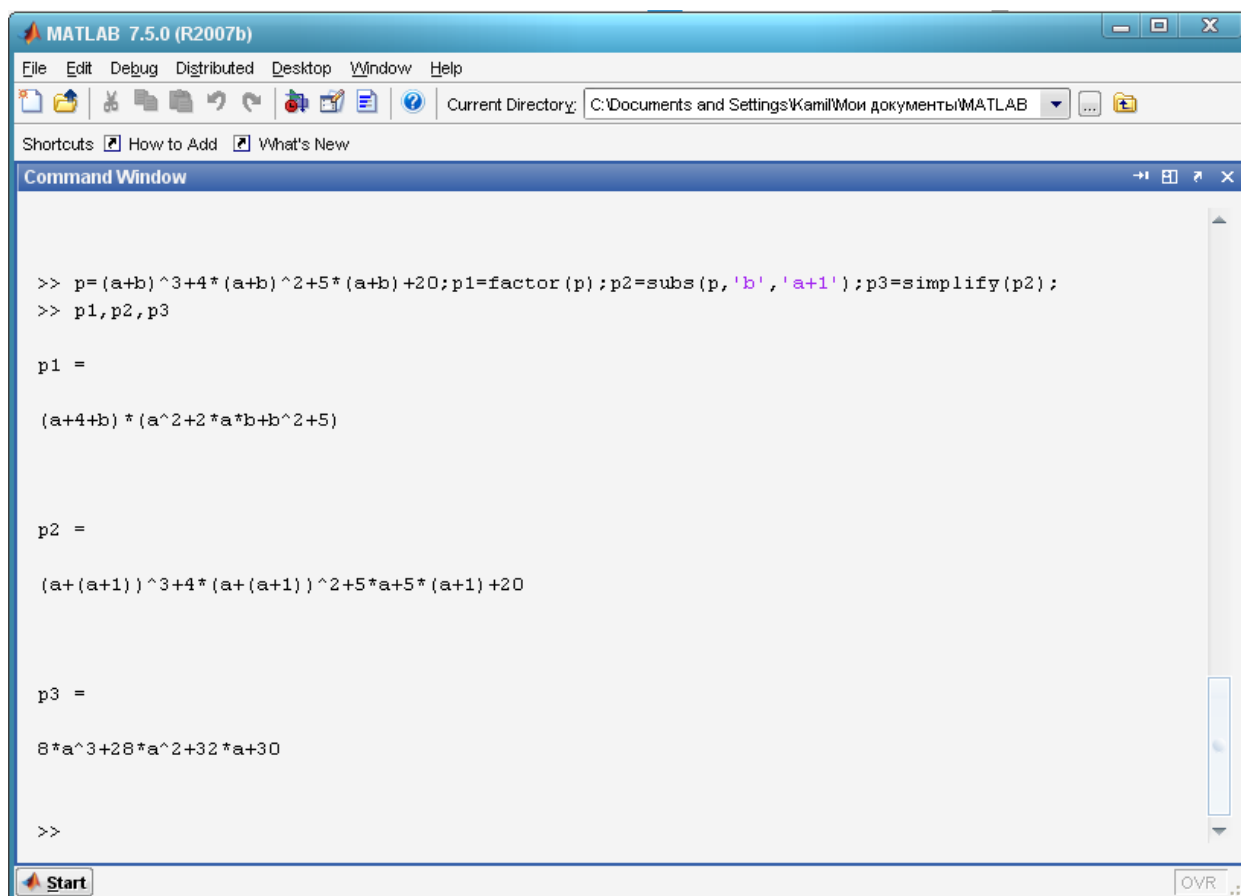
`p = (a + b)^3 + 4 * (a + b)^2 + 5 * (a + b) + 1 ;`

`p=factor(p)`

`p1=subs(p, 'b', 'a+1')`

`p2=simplify(p1)`

Natija:



5.3 - rasm. O'rniga qo'yish va soddalashtirish.

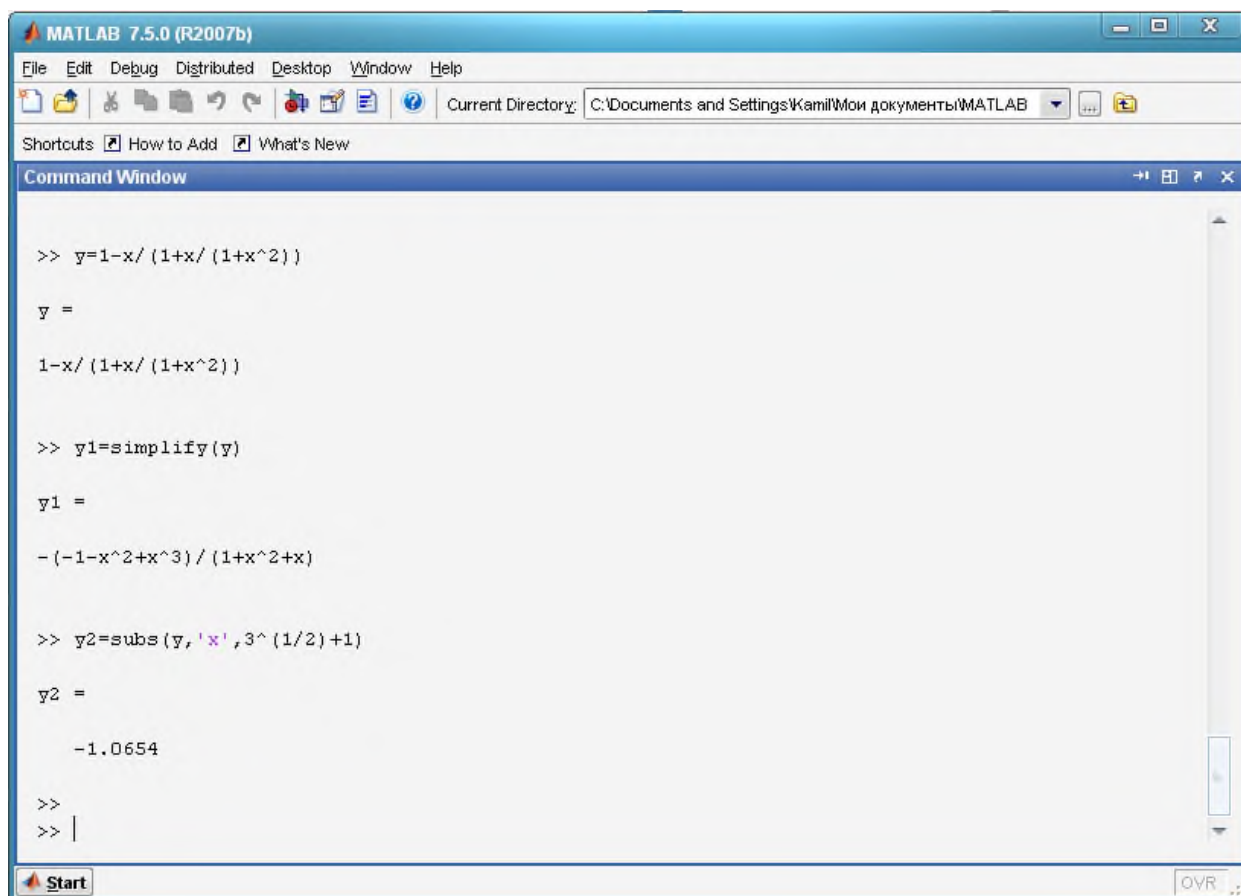
Yuqorida keltirilgan komandalardan foydalanib, murakkab ifodalarni qiymatlarini ham hisoblash mumkin. Masalan, $y = 1 - \frac{x}{1 + \frac{x}{1 + x^2}}$ ifodani

soddalashtirish va $x = \sqrt{3} + 1$ da qiymatini hisoblash kerak bo'lsin. Bu xolda quyidagi komandalar ketma-ketligi etarli:

```

>> syms x
>> y=1-x/(1+x/(1+x^2))
>> y=simplify(y)
>> y=subs(y,'x',3^(1/2)+1)

```



5.4 - rasm. Murakkab ifodani soddalashtirish.

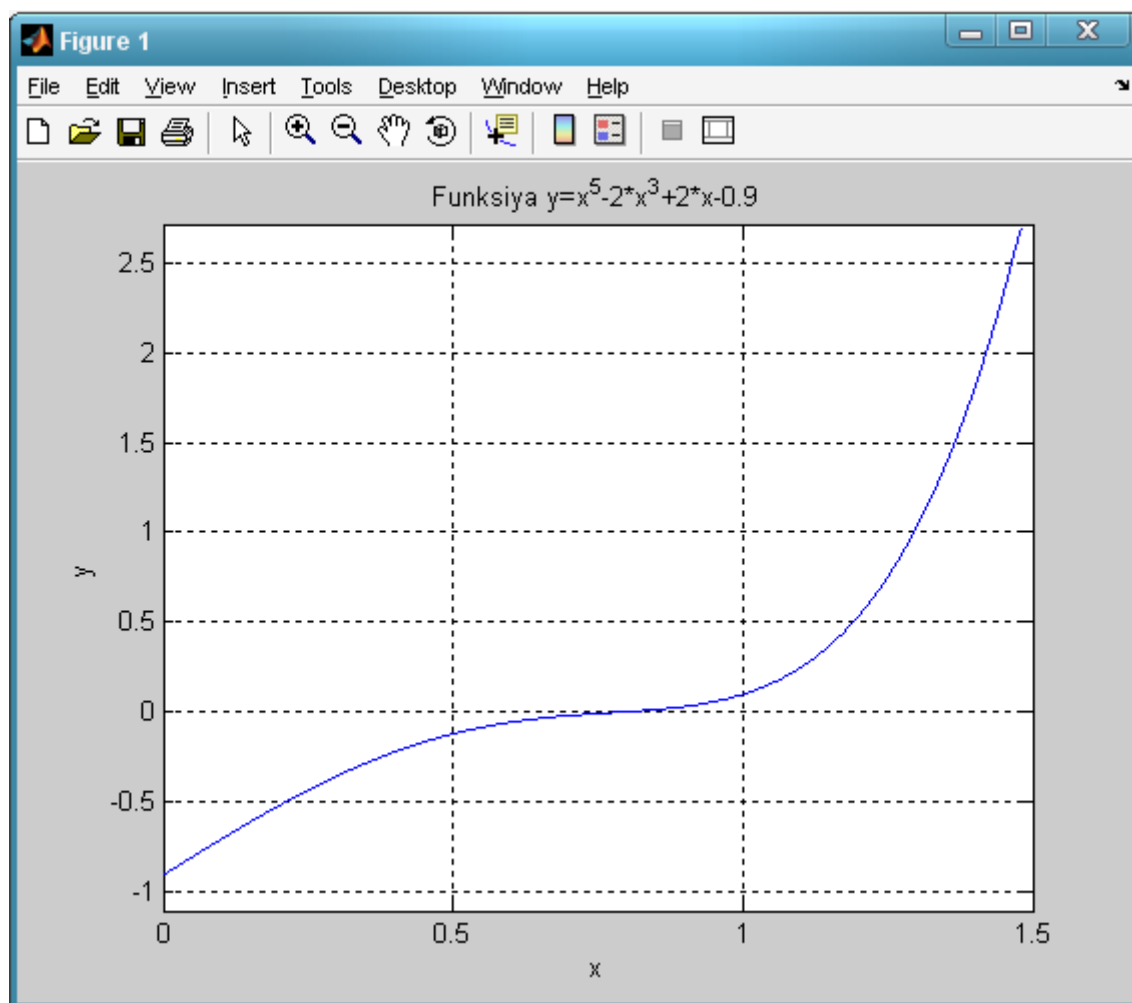
5.2. Simvulli o'zgaruvchilar yordamida algebraik tenglamalarni yechish

Matlab tizimida simvulli o'zgaruvchilar yordamida grafik chizish va algebraik tenglamalarni echish imkoniyati mavjuddir. Yechimni grafik usulda topish uchun ezplot funksiyasidan foydalaniladi. Misol uchun $y=x^5-2x^3+2x-0,2$ polinomni ildizlarini topishga xarakter qilaylik. Buning uchun simvulli o'zgaruvchilardan foydalanib ezplot(y) yordamida grafik quramiz va funksiya noli joylashgan oraliqni taxminan aniqlaymiz. Bizning misolda bu oraliq $[0;1,5]$ bo'ladi. Yechimni aniqlash uchun quyidagi komandalar ketma-ketligini yozamiz va grafikni hosil qilamiz(5.5-rasm):

```

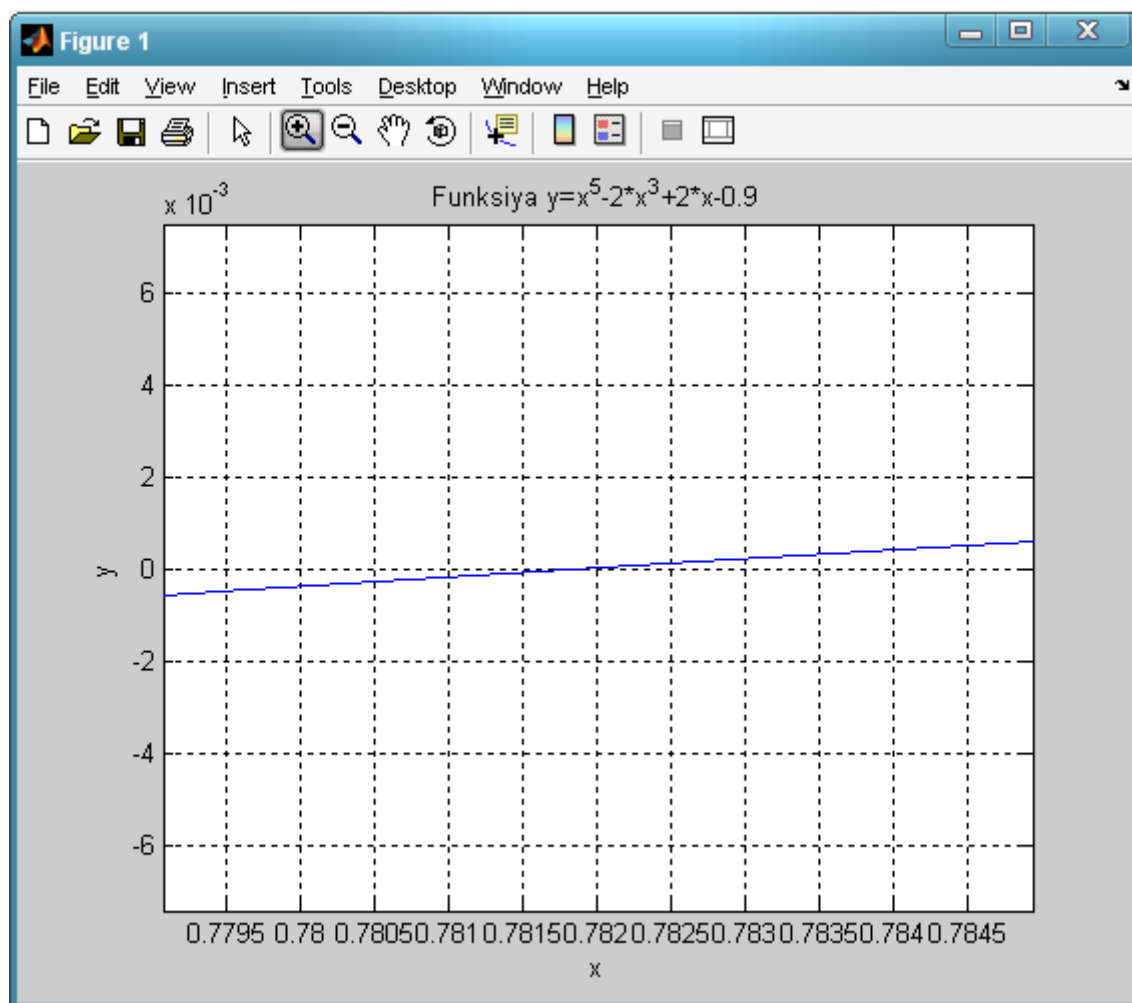
syms x y
y=x^5-2*x^3+2*x-0.2;
h=ezplot(y, [-1,1]);
grid on; ylabel('y'); xlabel('x');
title('Funksiya y=x^5-2*x^3+2*x-0.2')

```



5.5-rasm. Funksiyaning berilgan oraliqdagi grafigi.

Endi grafik oynada Zoom In knopkasini ishlatib, grafikni masshtablaymiz va kerakli aniqlikdagi echimni aniqlaymiz. Grafikni masshtablashda uni OX o'qini taxminan kesib o'tayotgan nuqtada bajarishimiz lozim bo'ladi. Kerakli aniqlikka erishish uchun masshtablash bir necha marta bajarilishi mumkin. Masshtablashni 5.5 -rasmdagi grafikda bir necha marta bajarib, quyidagini olamiz(5.6-rasm):



5.6-rasm. Masshtablangan grafik.

Tenglamani echish uchun Matlabning sozlangan funksiyasi solve dan ham foydalanish mumkin. Bu funksiya echimni analitik formada topib beradi. Undan keyin esa, echimni ko'rsatilgan aniqlikda ifodalab beruvchi vpa(y,n) (n-verguldand keyingi belgilar soni) funksiyasini qo'llash kerak.

Agar tenglama to'rt va undan yuqori tartibli, irratsional yoki transendent bo'lsa, solve funksiyasi echimni taqribiy sonli qiymatini aniqlab beradi. Yuqoridagi tenglamada ham xuddi shunday echimlar aniqlangan.

Endi masalani quyidagicha qo'yamiz: $x^5 - 2x^3 + 2x - 0.9 = 0$ tenglama echimini simvolli o'zgaruvchilar yordamida solve funksiyasini qo'llab toping va argumentning shu qiymatida $y = x^5 - 2x^3 + 2x - 0.9$ polinom qiymatini ham aniqlang.

```
>> syms x y; y=x^5-2*x^3+2*x-0.9; x=solve(y,x)
```



```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\Kamil\Мои документы\MATLAB
Shortcuts How to Add What's New
Command Window
>> syms x y; y=x^5-2*x^3+2*x-0.9; x=solve(y,x)

x =

    .78192805047996738188931242803535
    .79400127847908768950830246532206+.21926580439439730884763366017140*i
   -1.1849653037190713804529586793397+.54056363763338350223786389502194*i
   -1.1849653037190713804529586793397-.54056363763338350223786389502194*i
    .79400127847908768950830246532206-.21926580439439730884763366017140*i

>> x=vpa(x,3)

x =

    .782
    .794+.219*i
   -1.18+.541*i
   -1.18-.541*i
    .794-.219*i

```

5.7-rasm. Solve funksiyasidan foydalanish.

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\Kamil\Мои документы\MATLAB
Shortcuts How to Add What's New
Command Window
>> x=[.101; 1.07+.428*i; -1.12+.478*i; -1.12-.478*i; 1.07-.428*i];
>> y1=subs(y,'x',x(1));
>> y2=subs(y,'x',x(2));
>> y3=subs(y,'x',x(3));
>> y4=subs(y,'x',x(4));
>> y5=subs(y,'x',x(5));
>> y=[y1 y2 y3 y4 y5];
>> y=vpa(y,4)

y =

[    -.7000,   -.6960-.5465e-2*i,   -.7102-.7521e-2*i,   -.7102+.7521e-2*i,   -.6960+.5465e-2*i,

>>
>>
>>

```

5.8 - rasm. O'rniga qo'yib tekshirish.

Algebraik tenglamalarni echish uchun Matlab tizimida yana boshqa sozlangan funksiyalar ham bor. Ular quyidagilardan iborat:

1) **[x, f]=fzero('F',x0)** – x echimni va shu nuqtadagi funksiya qiymatini chiqaradi; bu erda F – tenglama chap tomonini qiymatini baholovchi fayl funksiyaning nomi yoki tenglama chap tomoni bo'lishi mumkin, x0 esa [a b] vektor yoki [a ,b] oraliqqa tegishli son(boshlanqich nuqta), $F(a)*F(b)<0$;

2) **[x, f]=fsolve('F',x1)** – bu erda x echim, f esa shu nuqtadagi funksiya qiymati, x1 – boshlanq'ich nuqtalardan tuzilgan massiv.

3) **R=roots(a)** – p polinomning ildizlarini taqribiy qiymatlarini beradi (a – polinom koeffisientlari va ozod hadidan tuzilgan vektor).

Misol. Ushbu tenglamani eching: $-x^2/200+5\sin x/x = 0$

Buning uchun avval chap tomonda turgan funksiyaning grafigini chizamiz va solve funksiyani qo'llaymiz:

```
>>clear
```

```
>>syms x y
```

```
>>u=-x^2/200+5*sins(x)/x;
```

```
>>ezplot(y,[-20,20]) ; grid on % koordinata tekisligiga to'r chizadi
```

```
>>x=solve(y,x) ; hold on % grafik oynani ochiq holda ushlab turadi
```

U hxolda quyidagi natijani olamiz:

```
x=
```

```
[empty sym ]
```

ya'ni solve funksiyasi bo'sh simvol massivini - "echim yo'q " degan ma'lumotni berayapti. Endi boshqacharoq yo'l tutamiz. Grafikdan foydalanib(5.9 - rasm), boshlanq'ich nuqtalarni tanlab olamiz va fsolve funksiyasi yordamida echimlarni topamiz:

```
>>syms x y
```

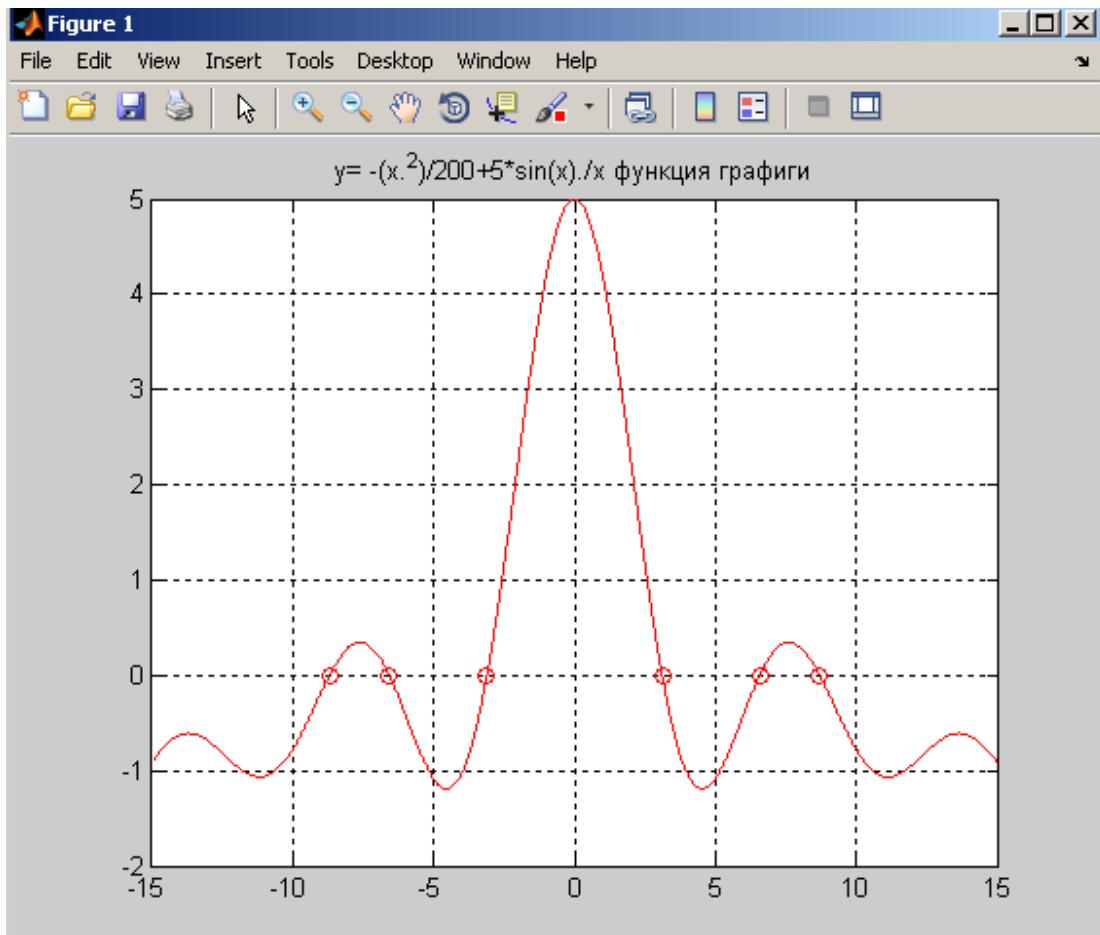
```
>>[x,y]=fsolve('-0.005*x.^2+5*sin(x.)/x' ,[-8 -7 -3 3 7 8 ])
```

```
>>plot( x,y , 'ro') % echim nuqtalarni qizil (red) aylanachalar bilan chiqaradi (5.9 - rasm).
```

```
x= -8.7046 -6.5708 -3.1115 3.1115 6.5708 8.7046
```

```
y=1.0e-010*
```

```
-0.0005 -0.5524 0.0000 0.0000 -0.5456 -0.0007.
```



5.9 - rasm. Yechim oraliqlarini aniqlash grafigi.

Yuqorida keltirilgan misollardan ko'rinib turibdiki, `ezplot(y)`, `ezplot(y, [x1, x2])` komandalari `fsolve`, `fzero`, `solve` funksiyalari bilan birgalikda ishlatilsa, echimni aniqlash jarayoni universal bo'ladi. Ta'kidlash joizki, `fsolve` va `solve` funksiyalari chiziqli bo'lmagan tenglamalar sistemalarini echishda ham qo'llaniladi. Masalan, quyidagi tenglamalar sistemasini echish kerak bo'lsin:

$$\begin{cases} \sin^2 x + \sin^2 y = \frac{7}{4} \\ x + y = \frac{5\pi}{6} \end{cases}$$

Avval `syms x y` deb e'lon qilib, keyin quyidagilarni kiritamiz:

```
Command Window
z1 =
sin(x)^2+sin(y)^2-7/4

>> z2=x+y-5*pi/6

z2 =
x+y-5/6*pi

>> [a,b]=solve(z1,z2)

a =

1/3*pi
4/3*pi
3/2*pi
1/2*pi

b =

1/2*pi
-1/2*pi
-2/3*pi
1/3*pi
```

5.10-rasm. Nochiziqli tenglamalar sistemasini echish.

Nazorat savollari

1. Simvulli o'zgaruvchilar qanday e'lon qilinadi?
2. Ifodani yoyish uchun qanday komanda qo'llash kerak?
3. Qanday qilib o'zgaruvchiga qiymat berish mumkin?
4. Bir nechta parametrlarni almashtirish uchun nima qilish kerak?
5. Qaysi komanda funksiya nolini grafik usulda topadi?
6. Tenglama echimini bitta boshlanq'ich nuqta holida qanday komanda orqali topish mumkin?
7. Bir nechta boshlanq'ich nuqtalar berilganda –chi?
8. Nima uchun bir nechta komandalarni birgalikda ishlatish kerak?
9. Qaysi funksiyalarni tenglamalar sistemasiga qo'llash mumkin?
10. Funksiya nolini topuvchi komandalarni keltiring.

Mustaqil ishlash uchun misollar

1) $y = 10x^3 - 3x^2 - 2x + \frac{1}{2}$ funksiyaning nollarini fzero

funksiya orqali toping. Ildizlarning va ularga mos funksiya qiymatlarining o'rtta arifmetigini hisoblang.

2) $S = 10x^3 - 3x^2 - 2x + \frac{1}{2}$ funksiyaning nollarini f zero

funksiyasi orqali aniqlang (boshlanq'ich nuqtalar sifatida -1 va 0 ni qarang). Ildizlar va ularga mos funksiya qiymatlarining mos o'rtta geometrigini chiqaring.

3) $y = 7x^3 - 5x^2 - 3x + \frac{1}{3}$ funksiyaning $y=0$ nglama uchun ildizlarini

fsolve funksiya orqali toping (boshlanq'ich nuqta sifatida 0 va 1 ni qarang). Ildizlar va ularga mos funksiya qiymatlarining kvadratlarini hisoblang.

4) $(x-1)^5 + (x+3)^5 = 243(x+1)$ tenglamaning echimini toping (boshlanq'ich nuqta sifatida -3, -1, 1 ni oling. Funksiya mos qiymatlarini hisoblang).

5) Ushbu $3 - 4\cos 4a - \cos 8a - 8\cos^4 2a$ ifodani ko'paytuvchilarga ajrating. $a = \frac{\pi}{4}$ va $a = \frac{\pi}{6}$ bo'lsa, ifodaning qiymatini hisoblang va natijalarining tafovutini aniqlang.

6) Ushbu $\operatorname{tg}^3 x - \operatorname{tg}^2 x - 3\operatorname{tg} x + 3$ trigonometrik ifodani ko'paytuvchilarga ajrating.

7) Agar $x = \frac{\pi}{6}$ bo'lsa, uning qiymatini toping. Natijaning kvadrat ildizini bank formatida chiqaring.

8) $(x+3)^4 + (x+5)^4 = 16$ tenglamani eching. Ildizlarning yiq'indisini uzun formatda chiqaring. Boshlanq'ich nuqta sifatida -3, -5, 0, 4 ni oling.

9) $10x^3 - 3x^2 - 2x + 1 = 0$ ning echimini fzero funksiya yordamida toping (boshlanq'ich nuqtalar sifatida 0 va 1 ni oling). Yechimlarga mos funksiya qiymatlarini aniqlang va ularning o'rtta arifmetigini toping.

10) $S(x) = 27x^3 + 9x^2 - 48x + 20 = 0$ funksiyaning nollarini toping.

Yechimlarga mos funksiya qiymatlarni aniqlang. Ularning o'rtta geometrigini hisoblang. Boshlanq'ich nuqta sifatida -1, 0, 1 ni oling.

11) $4x^4 - 16x^3 + 3x^2 + 4x - 1 = 0$ tenglamani echimlarini toping. Chap tomondagi ifodaning ildizlarga mos qiymatlarini aniqlang.

6. KO'PHADLAR BILAN ISHLASH

Ko'phadlar (darajali ko'phadlar ham deyiladi) – matematik hisoblashlar va ma'lumotlarni qayta ishlashning keng qo'llaniladigan obyektidir. Ma'lumki, n-darajali ko'phadlar quyidagi ko'rinishda yoziladi:

$$P(x) = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1}.$$

Biz quyida ko'phadlar bilan ishlovchi Matlabning asosiy funksiyalari bilan tanishamiz. Matlabda ko'phadlar asosan ularning koeffitsiyentlaridan tuzilgan vektorlar bilan beriladi.

6.1. Ko'phadlar bilan boq'liq amallar

Matlab ko'phadni darajalari kamayib boruvchi had koeffitsiyentlaridan iborat vektor-qator kabi tasvirlaydi. Ko'phadlar bilan bog'liq amallar quyidagilar

Funksiya	Tavsifi
conv	Ko'phadlarni ko'paytirish.
deconv	Ko'phadlarni bo'lish.
poly	Berilgan ildizlari orqali ko'phadni topish.
polyder	Ko'phadning hosilasini topish.
polyfit	Ko'phad ko'rinishida berilgan ma'lumotlarni approksimatsiya qilish.
polyval	Ko'phadning berilgan nuqtalardagi qiymatini topish.
polyvalm	Matritsali ko'phadning qiymatini hisoblash.
residue	Oddiy kasrlarga yoyish (chegirmalarni hisoblash).
roots	Ko'phad ildizlarini hisoblash.

Masalan, quyidagi misolni ko'rib chiqamiz: $p(x) = x^3 - 2x - 5$. Bu Vallis (Wallis) ning Fransuz Akademiyasida Nyuton usulini birinchi taqdim etishidagi mashhur misolidir. Ushbu misoldan keyinchalik turli Matlab funksiyalarini qo'llashda polinom sifatida foydalanamiz. Berilgan ko'phadni Matlab ga kiritish uchun quyidagini yozish kerak: $p = [1 \ 0 \ -2 \ -5]$.

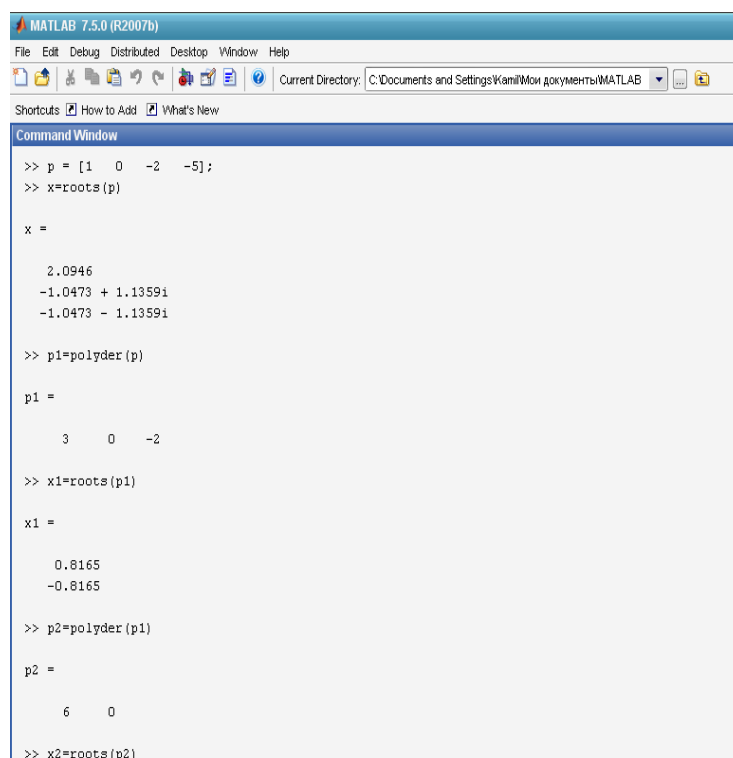
6.2. Ko'phadlarning qiymatlarini va ildizlarini hisoblash

Ko'phadning ildizlari roots funksiyasi yordamida hisoblanadi. Masalan,

```
>> p=[1 0 -2 5]; x = roots(p)
x = 2.0946
-1.0473 + 1.1359i
-1.0473 - 1.1359i
```

Ko'phad koeffitsientlarini roots funksiyasi argumentiga vektor shakldagi ko'rinishini ham qo'ysa bo'ladi. Masalan: >> r=roots([1 3 5 7])

```
r = -2.1795
-0.4102 + 1.7445 i
-0.4102 - 1.7445 i
```



```
MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\Kamil\Мои документы\MATLAB
Shortcuts How to Add What's New

Command Window

>> p = [1 0 -2 -5];
>> x=roots(p)

x =

    2.0946
   -1.0473 + 1.1359i
   -1.0473 - 1.1359i

>> p1=polyder(p)

p1 =

     3     0    -2

>> x1=roots(p1)

x1 =

    0.8165
   -0.8165

>> p2=polyder(p1)

p2 =

     6     0

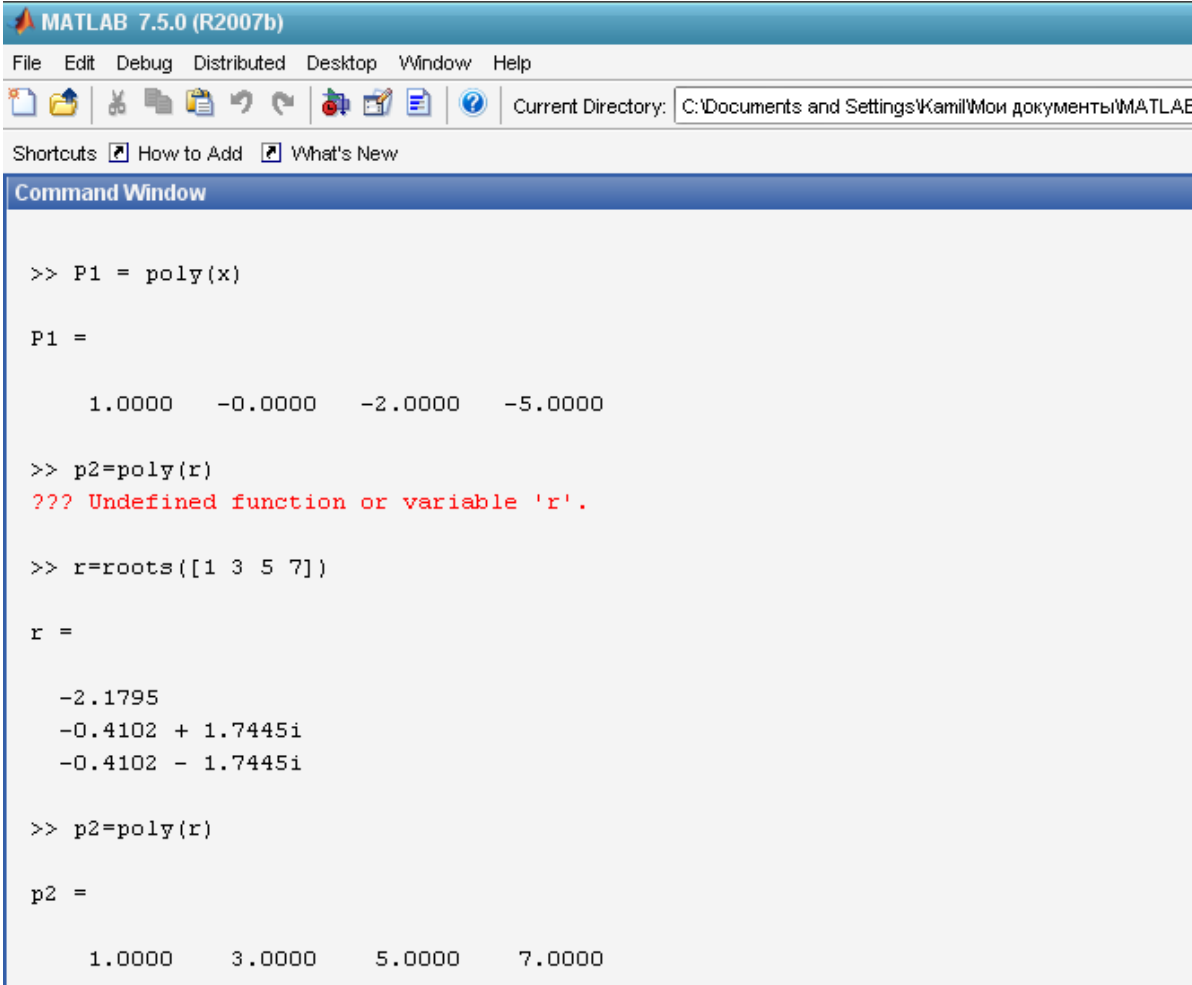
>> x2=roots(p2)
```

6.1 - rasm. Ko'phadning ildizlarini topish.

Matlab hisoblangan ildizlarni vektor-ustun ko'rinishida eslab qoladi. poly funksiyasi teskari amalni bajaradi, yani berilgan ko'phadning ildizlari bo'yicha uning koeffitsiyentlarini hisoblaydi. Bu funksiyani yuqoridagi polinom ildizlariga qo'llaymiz:

```
>>P1 = poly(x)
P1 = 1 0 -2 -5
>>poly(r)
```

ans = 1.0000 3.0000 5.0000 7.0000



```
MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\Kamil\Мои документы\MATLAB
Shortcuts How to Add What's New

Command Window

>> P1 = poly(x)

P1 =

    1.0000   -0.0000   -2.0000   -5.0000

>> p2=poly(r)
??? Undefined function or variable 'r'.

>> r=roots([1 3 5 7])

r =

   -2.1795
  -0.4102 + 1.7445i
  -0.4102 - 1.7445i

>> p2=poly(r)

p2 =

    1.0000    3.0000    5.0000    7.0000
```

6.2 – rasm. Ko'phadlarni ildizlar bo'yicha tiklash.

Ko'phadning qiymatlarini hisoblash uchun polyval funksiyasidan foydalaniladi. Bu funksiya ko'phadning qiymatini berilgan nuqtalarda hisoblaydi. Masalan, $x = 5$ nuqtada p ko'phadning qiymatini hisoblash uchun quyidagicha yoziladi:

```
polyval(p,x)
ans = 110
```

Shuningdek, matritsali ko'phadning qiymatini ham hisoblash mumkin. Vallis ko'phadining o'rniga quyidagi ko'phadni yozish mumkin: $p(X)=X^3-2*X-5*I$

bu erda X -kvadrat matritsa, I - birlik matritsa. Masalan, quyidagi X kvadrat matritsani shakllantiramiz: $X = [2 \ 4 \ 5; -1 \ 0 \ 3; 7 \ 1 \ 5]$; va yuqorida berilgan $p(X)$ ko'phadning qiymatini ushbu matritsada hisoblaymiz.

```
>> y = polyvalm(p, X)
```

```
y =    111    81   136
```


490 253 639
377 179 439

Shunga o'xshash quyidagi hisoblashlarni ham amalga oshirish mumkin:

```
Command Window

>> pp=polyval(p,p2)

pp =

    -6.0000    16.0000   110.0000   324.0000

>> pp=polyvalm(p,p2)
??? Error using ==> polyvalm at 29
Matrix must be square.

>> p8=[p;pp;p2;p2+7]

p8 =

    1.0000         0    -2.0000    -5.0000
   -6.0000   16.0000   110.0000   324.0000
    1.0000    3.0000    5.0000    7.0000
    8.0000   10.0000   12.0000   14.0000

>> pp=polyvalm(p,p8)

pp =

  1.0e+006 *

   -0.0005   -0.0020   -0.0075   -0.0197
    0.0700    0.1850    0.5727    1.4167
    0.0085    0.0144    0.0299    0.0613
    0.0272    0.0441    0.0851    0.1664
```

6.3 - rasm. Ko'phadlarning qiymatlarini hisoblash.

Yuqoridagi (6.3-rasm) xatolik shuni ko'rsatadiki, polyvalm(p, X) funksiya X ni o'rnida faqat kvadrat matritsa bo'lsagina ishlar ekan.

6.3 . Xarakteristik ko'phadlar

poly funksiyasi kvadrat matritsaning xarakteristik ko'phadi qiymatini ham hisoblaydi: >>A = [1.2 3 -0.9; 5 1.75 6; 9 0 1];

```
>>poly(A)
```

```
ans = 1.0000 -3.9500 -1.8500 -163.2750
```

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Documents and Settings\Kamil\Мои документы\MATLAB
Shortcuts How to Add What's New

Command Window

>> A = [1.2 3 -0.9; 5 1.75 6; 9 0 1];
poly(A)

ans =

    1.0000   -3.9500   -1.8500  -163.2750

>> roots(ans)

ans =

    7.2826
 -1.6663 + 4.4321i
 -1.6663 - 4.4321i

>> rr=ans

rr =

    7.2826
 -1.6663 + 4.4321i
 -1.6663 - 4.4321i

>> poly(rr)%характеристик kophadni qayta hosil qilib tekshirish

ans =

    1.0000   -3.9500   -1.8500  -163.2750

```

6.4 - rasm. Xarakteristik ko'phadni hosil qilish.

Ushbu ans ko'phadning **roots** funksiyasi yordamida hisoblangan ildizlari A matritsaning xususiy qiymatlari (xarakteristik sonlar) deyiladi.

6.4. Ko'phadlarni ko'paytirish va bo'lish

Ko'phadlarni ko'paytirish va bo'lish uchun mos ravishda **conv** va **deconv** funksiyalaridan foydalaniladi.

Quyidagi $a(s) = s^2 + 2s + 3$ va $b(s) = 4s^2 + 5s + 6$ ko'phadlarni ko'rib chiqamiz. Ularning ko'paytmasini topish quyidagicha amalga oshiriladi:

$$a = [1 \ 2 \ 3]; \quad b = [4 \ 5 \ 6];$$

$$c = \text{conv}(a,b)$$

Matlabda natija quyidagicha bo'ladi:

```
c =  
4 13 28 27 18
```

```
Misol: >> conv([1 2 3],[5 6])
```

```
ans =  
5 16 27 18
```

s ko'phadni b ko'phadga bo'lish uchun deconv funksiyasidan foydalanamiz:

```
[q,r] = deconv(c, b)
```

```
q =  
4 5 6
```

```
r =  
0 0 0 0 0 ,
```

bu erda r – bo'lishdan chiqqan qoldiq (bu holda nol). Umumiy holatda q, r, c, a ko'phadlar uchun deconv funksiyasida quyidagi munosabat o'rinli:

$$c = \text{conv}(q, a) + r$$

Misollar:

```
>> [c,r]=deconv([1 2 3],[5 6])
```

```
c = 0.2000 0.1600
```

```
r = 0 0 2.0400
```

```
>>p1=[2 0 1];% p1 va p2 ko'phadlarni kopaytiring
```

```
>>p2=[1 0 0 -1];
```

```
>>p=conv(p2,p1)
```

```
p = 2 0 1 -2 0 -1 % Solishtiring :(2x^2+1) (x^3-1) = 2x^5+x^3-2x^2-1
```

```
%Quyidagi ko'phadlarni boling: (2x^5+x^3-2x^2-1) / (x^3-1) = (2x^2+1)
```

```
>>deconv(p,p2)
```

```
ans = 2 0 1
```

6.5. Ko'phadlarning hosilasini hisoblash

polyder funksiyasi ixtiyoriy ko'phadning hosilasini hisoblaydi. Bu funksiyadan 6.2 punkt misolida foydalandik (6.1 -rasm.)

p = [1 0 -2 -5] ko'phadning hosilasini hisoblash quyidagicha bo'ladi:

```
q = polyder(p)
```

```
q =  
3 0 -2
```

polyder funksiyasi shuningdek ikki ko'phadning ko'paytmasi yoki bo'linmasining hosilasini ham hisoblaydi. Masalan, ikki **a** va **b** ko'phadni kiritamiz:

```
a = [1 3 5];    b = [2 4 6];
```

$a*b$ ko'paytmaning hosilasini hisoblash uchun ikkita kirish va bitta chiqish argumentli **polyder** funksiyasini kiritamiz:

```
c = polyder(a, b)
```

```
c =
```

```
8 30 56 38
```

a/b bo'linmaning hosilasini hisoblash uchun ikkita chiqish argumentli **polyder** funksiyasini kiritamiz:

```
[q, d] = polyder(a, b)
```

```
q =
```

```
-2 -8 -2
```

```
d =
```

```
4 16 40 48 36,
```

bu yerda q/d munosabat ikki ko'phadning differensiyalash natijasi (bu erda q -suratda hosil bo'lgan ko'phad koeffitsiyentlari, d - maxrajda hosil bo'lgan ko'phad koeffitsiyentlari).

Misollar:

```
>>polyder([1 -2 3 4 5])
```

```
ans =
```

```
4 -6 6 4
```

```
>> polyder([1 2 3],[5 6])
```

```
ans =
```

```
15 32 27
```

```
>> [c,r]=deconv([1 2 3],[5 6])
```

```
c = 0.2000 0.1600
```

```
r = 0 0 2.0400
```

```
>>p1=[2 1 0 -1 0 -3];
```

```
>>polyder(p1);
```

```
% p1 ning hosilasi
```

```
>>p2=[1 0 0 -1];
```

```
>>polyder(p1,p2);
```

```
% (p1*p2) ning hosilasi
```

```
>> [q, r]=polyder(p1,p2)
```

```
% (p1/p2) ning hosilasi
```

```
q =
```

```
4 1 0 -9 -4 9 2 0
```

r =

1 0 0 - 2 0 0 1

Nazorat savollari

1. Ko'phadning ta'rifini ayting.
2. Ko'phad tartibi nima?
3. Ko'phad ildizi deganda nima tushuniladi?
4. roots komandasini tushuntiring
5. Qanday funksiya ildizlarga asoslanib ko'phadni tiklaydi?
6. O'zgaruvchining berilgan qiymatida ko'phad qiymatini qanday topish mumkin?
7. O'zgaruvchi matritsa bo'lishi mumkinmi?
8. Ko'phadlar ko'paytmasi va bo'linmasini topish uchun nima qilish kerak?
9. Qaysi funksiya yordamida ko'phad hosilasi topiladi?
10. Ko'paytma va bo'linma holda hosilani topish uchun qanday formatlar qo'llanishi kerak?

Mustaqil ishlash uchun misollar

Quyidagi ko'phadlarga barcha amallarni qo'llang:

- 1) $P = 15X^5 + 10X^4 - 2X^3 - 1$
- 2) $P = 13X^6 - 7X^3 + 2,5X^2 - 4$
- 3) $P = -27X^4 - 3,3X^5 + 2X^4 - 1,7X^3 + X^2$
- 4) $P = 2X^2 - X + 17$
- 5) $P = -1,7X^3 + 2X^2 - 4X + 3$
- 6) $P = 10X^{10} - 8X^8 + 6X^6 - 4X^4 + 2X^2 - 1$
- 7) $P = (a + 1)^5$
- 8) $P = (\sin X - 1)^6$
- 9) $P = (1 - X^2)^3$
- 10) $P = (5 - t^3)^7$

7. DASTURLASH ASOSLARI. MATLABDA MA'LUMOTLAR VA FAYLLARNING TOIFA(TIP)LARI

7.1. Matlabda dasturlash vositalari

Matlab tizimidagi dasturlar matn formatidagi m-fayllardir. Matlab tizimida dasturlash tili quyidagi vositalarga ega:

- Har xil turdagi ma'lumotlar;
- Konstantalar va o'zgaruvchilar;
- Operatorlar(matematik ifodalarning operatorlarini ham o'z ichiga oladi);
- Biriktirilgan komanda va funksiyalar;
- Foydalanuvchining funksiyalari;
- Boshqaruvchi strukturalar;
- Sistema operatorlari va funksiyalar;
- Dasturlash tilining kengaytirish vositalari.

Matlab tizimida dasturlash kodlari yuqori darajali tilda yoziladi va ushbu til tipik interpretator bo'lib hisoblanadi, yani dasturning har xil instruksiyasi darhol taniladi va bajariladi. Hamma instruksiyalarni, yani to'liq dasturni kompilyatsiya qilish etapi mavjud emas. Matlab bajariluvchi dasturlarni yaratmaydi. Dasturlar faqat m-fayllar ko'rinishida mavjud bo'ladi. Dasturlarning ishlashi uchun Matlab muhiti zarur. Lekin Matlabda yozilgan dasturlarni C va C++ dasturlash tillariga translyatsiya qiluvchi kompilyatorlar yaratilgan. Ular Matlab muhitida tayyorlangan dasturlarni bajariluvchi dasturlarga aylantirish masalasini hal qilish imkoniyatini beradi. Matlab tizimi uchun kompilyatorlar mustaqil dasturiy vositalardir.

Shuni esda tutish kerakki, Matlabning hamma instruksiyalari ham kompilyatsiya beravermaydi, yani kompilyatsiyadan oldin bunday dasturni qayta ishlash talab qilinadi. Kompilyatsiya qilish dasturlarning bajarilish tezligini 10-15 martagacha ortirishi mumkin.

7.2. Matlabda ma'lumotlar toifalari

Matlabda quyidagi toifadagi ma'lumotlardan foydalaniladi:

- sonli toifa;
- simvollar va qatorlar ;
- ob`yektlar (matrisalar);

Sonli toifada berilgan ma'lumot ikki xil - haqiqiy va kompleks sonlar bo'lishi mumkin. Haqiqiy sonlar huddi matematikadagi kabi ishlatiladi. Butun va kasr qismlari nuqta(.) bilan ajratiladi. Kompleks sonlar esa, avval eslatganimizdek, $a+ib$ yoki $a+bi$ ko'rinishida yoziladi, bu yerda a va b mos ravishda kompleks sonning haqiqiy va mavhum qismlari deyiladi, i -belgi (yoki j) mavhum birlikni bildiradi ($i^2=-1$). Kompleks sonni bildiruvchi i -belgi b ning chap yoki o'ng tomoniga probelsiz yozilishi kerak, aks holda Matlab tizimi xatolik haqida axborot beradi.

Umuman, ixtiyoriy toifadagi son matritsalarini, vektorlarni yoki skalyar miqdorlarni elementlari (qiymatlari) bo'lishi mumkin. Xotirada barcha sonlar ikki karrali aniqlikdagi son ko'rinishida saqlanadi. Sonlar aniqlangan oraliqlarning chegaralari hamda mashina aniqligi tizim o'zgaruvchilari `eps`, `realmax` va `realmin` orqali beriladi.

Matlabda apostroflar ichiga joylashtirilgan simvollar ketma-ketligi qator deb tushuniladi. Qatorlarga misol qilib quyidagilarni keltirish mumkin:

```
a='Matlab'
```

```
b='function'
```

Bir nechta qatorlarni birlashtirish uchun huddi vektor va matritsalar kabi [...] kvadrat qavslar ishlatiladi. Masalan,

```
str1=['This','is','string'],
```

```
str2=['Sistema','Matlab']
```

kabi ifodalar mos ravishda quyidagi simvolli qatorlarni beradi:

```
str1='This is string'
```

```
str2='Sistema Matlab'
```

Ob'ekt (matritsa)lar haqida yuqorida yetarlicha ma'lumotlar berilgan.

Qatorlarni hosil qiluvchi va ularga ishlov beruvchi Matlabning ba'zi funksiya(komanda)larini keltirib o'tamiz:

- `blanks(n)`- n ta probeldan iborat qatorni bildiradi;
- `num2str(n)`-haqiqiy sonni qatorga aylantiradi;
- `deblanks(s)`- s qatordan kerak bo'lmagan probellarni yo'qotadi;
- `index(s,t)`- s qatorda t qator ostining birinchi marta ko'rinishi holatini chiqaradi. Agar qator osti bo'lmasa, `nol`ni chiqaradi;
- `randex(s,t)`- s qatorda t qator ostining oxirgi marta ko'rinishi holatini chiqaradi. Agar qator osti bo'lmasa, `nol`ni chiqaradi;
- `strcmp(s1,s2)`- agar $s1$, $s2$ qatorlar bir xil bo'lsa, `1` ni chiqaradi, aks holda `0` ni chiqaradi;

- `strrep(s,x,y)`- x qator ostining s qatorga barcha kirishlarni y qatorga kirishga almashtiradi;
- `bin2dec(s)`- qator ko'inishida tasvirlangan ikkilik sistemasidagi songa mos o'nlik sistemasidagi sonni chiqaradi;
- `dec2bin(n)`- o'nli sistemadagi manfiy bo'lmagan songa mos ikkilik sistemasidagi sonni qator ko'inishida chiqaradi;

7.3. Fayllarning toifalari

Matlabda ikki xil toifadagi fayllardan foydalaniladi: ular fayl-ssenariy va fayl-funksiyalardir. Har bir faylni hosil qilinishini va xossalarini alohida-alohida ko'rib chiqamiz.

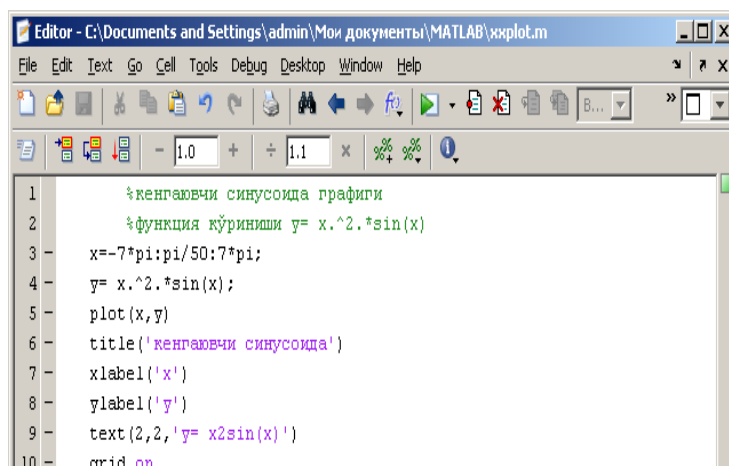
Shunday masalalar borki, ularni echish uchun bir nechta komandalar yoki komandalar qatorlarini, ularni bajarishdan avval yozishga to'q'ri keladi. Bunday masalalarni hal qilish uchun Matlabda m-fayllardan foydalaniladi. Buning uchun yangi m-faylda Matlabning bir nechta komandalari ketma-ketligi yoziladi va shu faylga nom berib ularni saqlab qo'yiladi. Natijada bu fayldagi komandalar ketma-ketligi Matlab komandalar oynasidan faylga murojaat qilish orqali bajarilishi mumkin. Mana shunday qo'shimcha hosil qilingan fayl ishchi fayl yoki fayl-ssenariy deyiladi. Bunday fayl nom berib saqlanayotganda tizim avtomatik ravishda uni nomiga *. m kengaytma beradi.

Demak, ishchi fayllar - Matlab komandalar ketma-ketligini o'z ichiga oluvchi oddiy m-fayllardir. Ishchi fayllar `matn(tekst)` tahririda va formatida tayyorlangan bo'lishi shart va Matlab yuklatilgan katalogda saqlangan bo'lishi kerak. Fayl nomi ixtiyoriy o'zgaruvchiga berish mumkin bo'lgan .m kengaytmali nom bo'ladi. Ishchi m-fayl yaratishga doir misol ko'raylik: $y=x^2\sin(x)$, $x\in[-7\pi; 7\pi]$ funksiyaning grafigi chizilsin. Buning uchun ishchi m-fayldan foydalanamiz. Yangi m-fayl chaqiramiz va unda Matlabning matnli tahrir va formatida quyidagicha komandalar ketma-ketligini kiritamiz:

```
% kengayuvchi sinusoida grafigi
% funksiya ko'inishi y= x2sin(x)
x=-7*pi:pi/50:7*pi;
y= x2sin(x);
plot(x,y),...
title('kengayuvchi sinusoida'),...
xlabel('x'),...
ylabel('y'),...
```



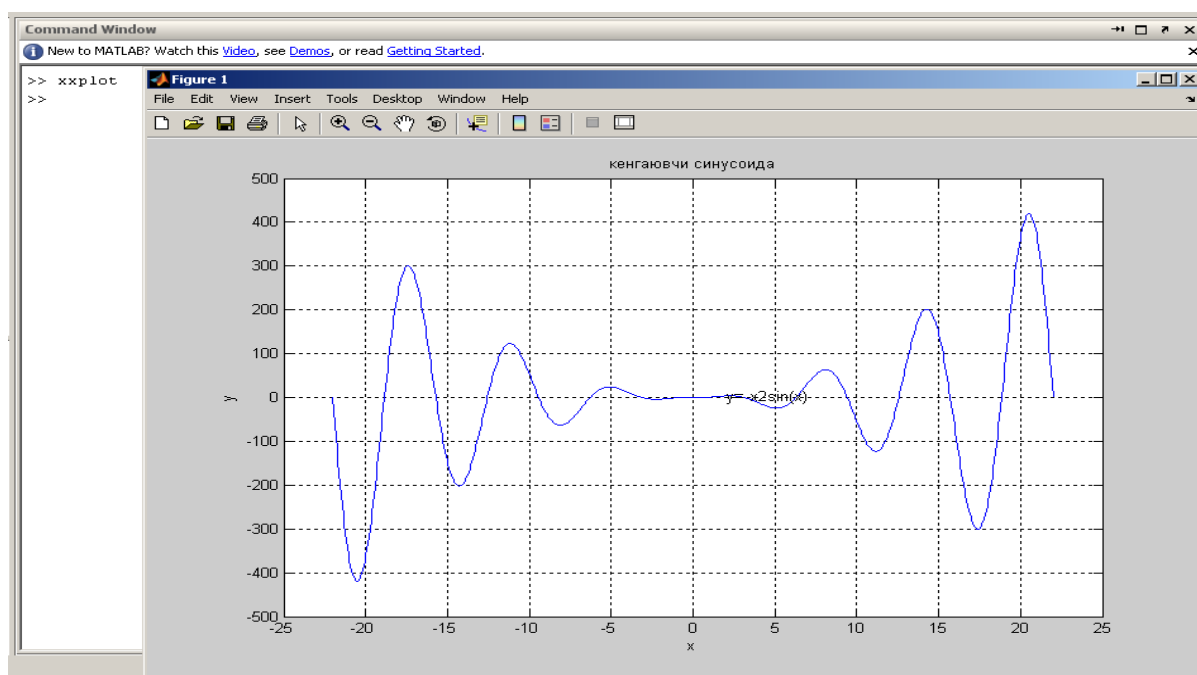
```
text(2,2,'y= x2sin(x)'),...  
grid on
```



```
Editor - C:\Documents and Settings\admin\Мои документы\MATLAB\xxplot.m  
File Edit Text Go Cell Tools Debug Desktop Window Help  
1 %кенгаювчи синусоида графиги  
2 %функция кўриниши y= x.^2.*sin(x)  
3 x=-7*pi:pi/50:7*pi;  
4 y= x.^2.*sin(x);  
5 plot(x,y)  
6 title('кенгаювчи синусоида')  
7 xlabel('x')  
8 ylabel('y')  
9 text(2,2,'y= x2sin(x)')  
10 grid on
```

7.1-rasm. Fayl-ssenariy.

Endi yuqoridagi komandalar ketma-ketligi yozilgan fayl(7.1-rasm) ,masalan, xxplot.m nomi bilan Matlabning ishchi katalogida saqlab qo'yilishi kerak. Biz Matlab buyruqlar oynasidan xxplot komandasini kirgizib, kengayuvchi sinusoidaning grafigini olsak bo'ladi. Natija grafik oynada chiqadi:



7.2 - rasm. Kengayuvchi sinusoidaning grafigi.

Bu erda birinchi ikkita komanda % belgi bilan boshlangani uchun Matlab tizimi ularni matnli sharh sifatida qabul qiladi. Matlabda % belgidan keyin yozilgan ixtiyoriy komanda yoki matn sharh deb qabul

qilinadi va bajarilmaydi. Misollardagi boshqa komandalar tarifini kelgusi mavzularimizda keltiramiz.

7.4. Ssenariy fayllari (Script-fayl) tuzilishi va xossalari

Matlab tizimida dasturlashni komandalar rejimida amalga oshirish noqulay, chunki bu holda har bir qatordagi kamchilik uni qaytadan yozilishiga sabab bo'ladi. Matlab tizimida dasturlarning tashqi atributi bo'lib, m-faylda yozilgan amallarning ketma-ketligi hisoblanadi. Matlabda m-faylni yaratish uchun biriktirilgan tahrirlagichdan yoki ASSII formatini qo'llaydigan har qanday matn tahrirlagichdan foydalanish mumkin. Tayyorlangan va diskka yozilgan m-fayl Matlab tizimining bir qismiga aylanadi va uni komandalar oynasidan yoki boshqa m-fayldan chaqirish mumkin. Ikkala turdagi m-fayllarni (fayl- ssenariyalar va fayl- funksiyalar) ham tuzish jarayonida, ular Matlab tizimiga biriktirilgan m-fayllarning tahrirlagich/sozlagichi yordamida sintaksis nazoratdan o'tgan bo'lishi kerak.

Script-fayl deb ataluvchi fayl-ssenariyalar kirish va chiqish parametrlari bo'lmagan bir nechta komandalar qatorining to'plamidir. Ular quyidagi tarkibga ega bo'ladilar:

- %Asosiy izoh;
- %Qo'shimcha izoh;
- Bir nechta komandalarni o'z ichiga oluvchi faylning qobiq'i.

Fayl-ssenariy quyidagi xossalarga ega bo'ladi:

- Kirish va chiqish argumentlari bo'lmaydi;
- Ishchi sohadagi ma'lumotlar bilan ham ishlaydi;
- Bajarilish vaqtida kompilyatsiya bo'lmaydi;
- Fayl ko'rinishga keltirilgan, sessiyadagiga o'xshash amallar ketma-ketligidan iborat bo'ladi.

Matnli izohning birinchi satri asosiy izoh va keyingi satrlari qo'shimcha izoh bo'lib hisoblanadi. Asosiy izoh

```
lookfor <katalog_nomi> va help <katalog_nomi>
```

komandalari, to'liq izohlar esa

```
help <fayl_nomi>
```

komandasi bajarilganda ekranga chiqadi.

Quyidagi fayl-ssenariyni ko'raylik(7.3-rasm):

```
Editor - C:\Documents and Settings\admin\Мои документы\MATLAB\pqr.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 % qizil rangda sinusoidani ko'rinishi
2 % oraliq o'zgaruvchan holda berilgan
3 x=xmin:0.01:xmax;
4 plot(x,sin(x),'r')
5 title('oddiy sinusoida')
6 xlabel('x')
7 ylabel('y')
8 grid on
```

7.3 - rasm. Izohli fayl-ssenariy.

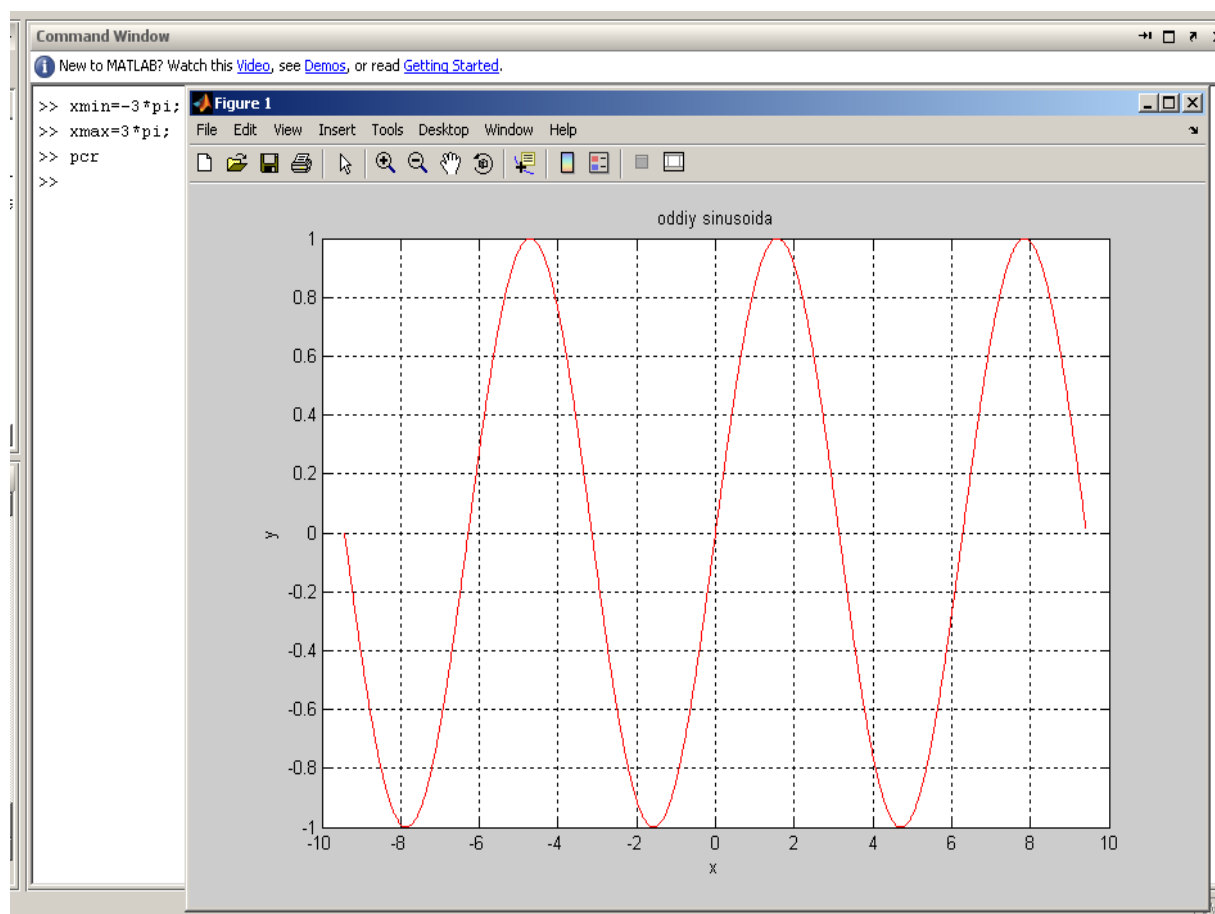
Dasturni pqr nomi bilan Matlab katalogida saqlaymiz va komandalar oynasida quyidagilarni kiritamiz va ENTER tugmasini bosamiz:

```
>> xmin=-3*pi;
```

```
>> xmax=3*pi;
```

```
>> pqr
```

Fayl -ssenariy ishga tushadi va ekranda quyidagicha tasvir hosil bo'ladi:



7.4 - rasm. Sinusoidaning grafigi.

Izohlarda % belgisi satrning birinchi pozitsiyasiga yozilishi kerak. Aks holda help name komandasi izohni qabul qilmaydi va No help comments found in-name.m ko'rinishidagi axborotni beradi.

Bunday faylni ishga tushirish uchun xmin va xmax o'zgaruvchilar oldindan tayyorlangan bo'lishi kerak. Fayl-ssenariyalarda ishlatiladigan o'zgaruvchilar global o'zgaruvchilar bo'lib hisoblanadi, yani ular sessiya komandalarida ham, dasturiy bloklarning (jumladam fayl-ssenariyalarning) ichida ham bir xil ishlaydi. Shuning uchun sessiyada berilgan qiymatlar faylda ishlatiladi. Fayl-ssenariylarning nomlaridan funksiyaning parametrlari sifatida foydalanish mumkin emas, chunki fayl-ssenariy qiymatlarni qaytarmaydi. Fayl-ssenariylarni kompelyatsiya qilib bo'lmaydi. Ular fayl-funksiyalarga aylantirilgandan keyingina kompelyatsiya qilinishi mumkin.

7.5. Fayl-funksiya va uning xossalari

Matlab tizimida foydalanuvchi uchun aniq bir maqsadli hisoblashlarni bajaruvchi va Matlab katalogida yo'q bo'lgan funksiya zarur bo'lib qoladi. Bunda foydalanuvchi yangi funksiyaning hosil qilib Matlab katalogiga qo'shib qo'yish imkoniyatiga ega. Yangi funksiyaning tashkil qiluvchi komanda va funksiyalar har doim matnli m-fayllarda joylashgan bo'ladi.

Yangi hosil qilingan, bir nechta komandalar ketma-ketligidan iborat funksiya o'zining nomiga, kirish parametrlari deb ataluvchi argumentlariga va lokal xarakterdagi o'zgaruvchilarga ega bo'lib, unga parametrlarga qiymat berish orqali nomi bilan murojat qilish mumkin.

Funksiya tuzib, saqlanayotgan m-faylning nomi alifbo belgilardan boshlanib *. m kengaytmasiga ega bo'ladi. Kengaytmasiz m-faylning nomi, bu Matlabda murojaat qilish mumkin bo'lgan fayl-funksiya yoki ishchi faylning nomidir.

Funksiya hosil qilinayotgan m-faylning boshlang'ich qatorlari matnli sharhlardan iborat bo'lib, shu funksiyaning mohiyatini, xossalari ochib beruvchi bo'lishi kerak. Undan keyingi birinchi qatorda aniqlangan funksiya nomi m-faylning kengaytmasiz nomi bilan bir xil bo'lishi kerak. Umumiy ko'rinishda m-fayldagi funksiya har doim function so'zidan boshlanib, quyidagicha bo'ladi:

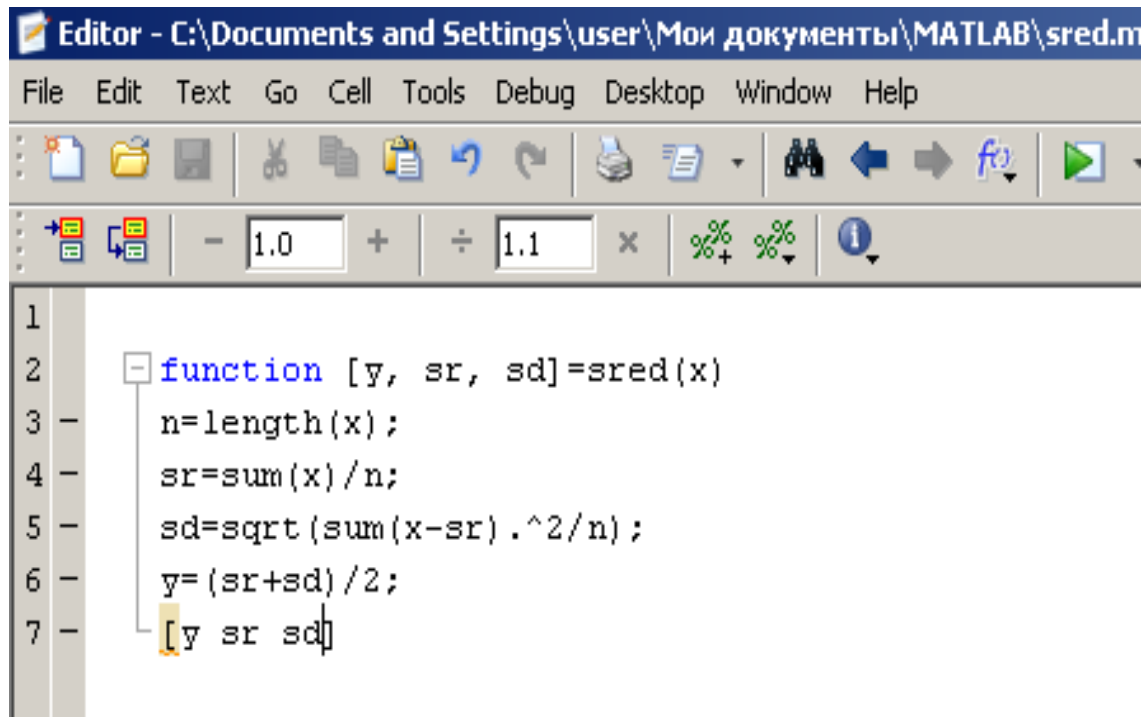
function y=<funksiya nomi>().

Funksiya nomidan keyin oddiy qavs ichiga argumentlar (parametrlar) vergul(,) bilan ajratib yoziladi. Masalan, diskdagi sred.m nomli fayldagi quyidagi

```
function [y, sr, sd]=sred(x)
n=length(x);
sr=sum(x)/n;
sd=sqrt(sum(x-sr).^2/n);
y=(sr+sd)/2;
```

kod sred nomi bilan aniqlangan fayl-funksiya x vektor koordinatalari o'rta arifmetigini (sr), standart chetlanishini (sd) hamda ularning o'rtasini (y) hisoblovchi yangi funksiyani aniqlaydi.

Funksiya ichidagi barcha o'zgaruvchilar lokal xarakterga egadir, sum(x) esa vektor koordinatalari yig'indisini hisoblovchi Matlab funksiyasidir.



7.5 - rasm. Izohsiz fayl-funksiya.

```

Command Window
>> sred(1:11)

ans =

     3     6     0

ans =

     3

>> sred(3:1.5:23)

ans =

 6.3750  12.7500     0

ans =

 6.3750

```

7.6 - rasm. Fayl-funksiyaning qo'llanilishi.

M-fayl funktsiya ichidagina ko'rinadigan funktsiya osti funktsiyasi ham bo'lishi mumkin. Bu funktsiya osti funktsiyasi ham asosiy fayl - funktsiya komandalardan keyin yozilib, u ham huddi asosiy fayl -funktsiya kabi aniqlanadi. Masalan, srg funktsiya sred fayldagi funktsiya osti bo'lsa, kod quyidagicha yozilishi mumkin:

```

function [y, sr, sd]=sred(x)
n=length(x);
sr=srg(x,n);
sd=sqrt(sum((x-srg(x,n)).^2)/n);
function sr=srg(x,n)
sr=sum(x)/n;

```

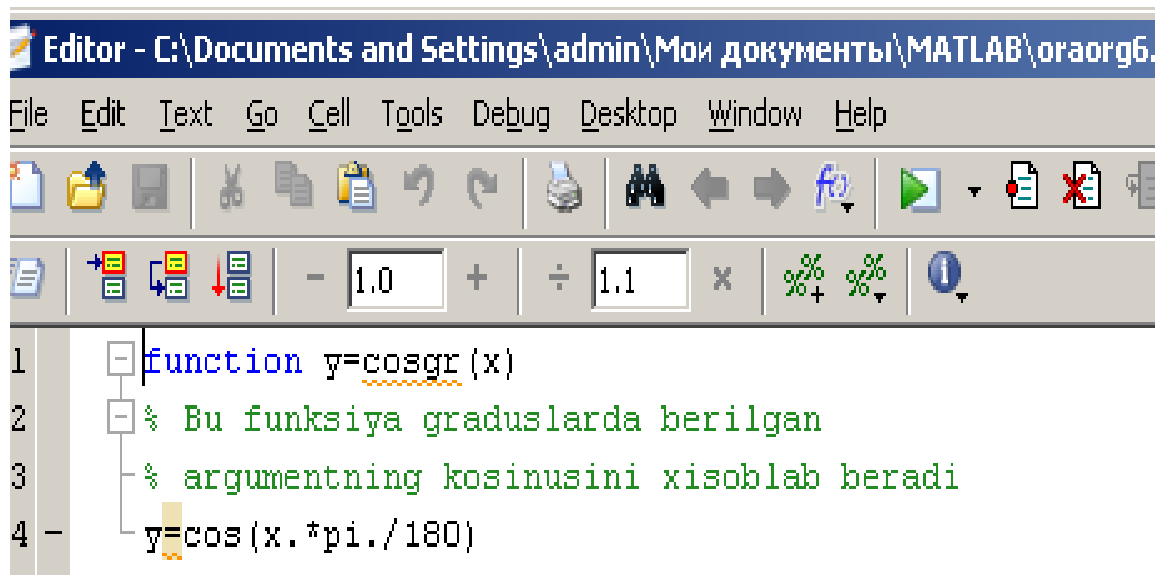
Agar Matlab tizimi funktsiyani nomi bo'yicha topa olmasa, u holda shu nomdagi faylni qidiradi. Funktsiya topilgandan keyin, uni keyinchalik ishlatish uchun Matlab tizimi funktsiyani xotiraga kompilyatsiya qiladi.

Funktsiya m-fayldan chaqirilsa, Matlab funktsiyani analiz qiladi va xotirada saqlab qo'yadi. Bu funktsiya xotira clear buyrug'i bilan tozalanmaguncha xotirada saqlanib turadi.

Matlab katalogidagi barcha trigonometrik funktsiyalar radian argumentlarda hisoblashni bajaradi. Endi biz graduslarda berilgan ixtiyoriy burchakning kosinusini hisoblab beruvchi fayl -funktsiya hosil qilish misolini ko'ramiz. Bunda shunga e'tibor berish kerakki, funktsiya nomi fayl nomi bilan bir xil bo'lishi kerak. Bu funktsiya uchun cosgr(x) ni

funksiya nomi qilib olaylik. Masalani hal qiluvchi kod quyidagicha bo'ladi:

```
Function y= cosgr(x)
% Bu funksiya graduslarda berilgan
% argumentning kosinusini hisoblab beradi
y=cos(x.*pi./180)
```



7.7 - rasm. Yangi tuzilgan fayl-funksiya.

Endi tizim ichida x ning aniq gradus qiymatlari bilan cosgr(x) ga murojat qilsak, unga qiymat chiqarib beradi:

```
>>cosgr(90)
ans=0
>>cosgr(180)
ans=-1
>>cosgr(45)
ans=0.7071.
```

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> cosgr (0)

z =

    1

ans =

    1

>> cosgr (1)

z =

    0.9998

ans =

    0.9998

>> cosgr (60)

z =

    0.5000
```

7.8 - rasm. Yangi fayl-funksiyaga murojat natijalari.

M-fayl funksiya quyidagi xossalarga ega bo'ladi:

- U function e'lon so'zi bilan boshlanadi, undan keyin o'zgaruvchining nomi va chiqish parametrlarning ro'yhati ko'rsatiladi;
- Funksiya o'z qiymatini qaytaradi va uni matematik ifodalarda nomi (parametrlar ro'yxati) ko'rinishida ishlatish mumkin;
- Fayl-funksiyaning qobig'idagi hamma o'zgaruvchilar lokal o'zgaruvchilardir, yani faqat funksiyaning ichida o'rinli;
- Fayl-funksiya mustaqil dasturiy modul bo'lib, boshqa modullar bilan o'zining kirish va chiqish parametrlari orqali aloqada bo'ladi;
- Fayl-funksiya Matlab tizimini kengaytirish vositasidir;
- Fayl-funksiya kompilyatsiya qilinadi va bajariladi, hosil qilingan mashina kodlari MATLAB tizimining ishchi sohasida saqlanadi.

7.6. Lokal va global o'zgaruvchilar

Funksiya parametrlarining ro'yhatida ko'rsatilgan parametrlar lokal o'zgaruvchilar bo'lib, funksiya chaqirilganda ularning o'rniga qo'yiladigan qiymatlarni olib o'tish uchun xizmat qiladi.

Agar funksiyadagi o'zgaruvchilar global bo'lishi zarur bo'lsa, ular global x1, x2, ... komandasi yordamida e'lon qilinadi.

Quyidagi misolni ko'raylik. Tahrirlagich oynasida $sws=(x+y+z)/abs(x+2*y+3*z)$ ifodani hisoblovchi uch o'zgaruvchili yu funksiyasi hosil qilingan.

Dasturda x,y va z o'zgaruvchilar yu(x,y,z) funksiyaning parametrlari bo'lganliklari sababli, ular lokal o'zgaruvchilardir. Funksiya qobiq'idan tashqarida ularga nol qiymatlar berilgan. Agar komandalar oynasida yu(1,2,1) ning qiymati hisoblanadigan bo'lsa, ularga x=1, y=2 va z=1 qiymatlar beriladi. Shuning uchun natija sws=0,5 bo'ladi. Lekin funksiyaning qobiq'idan chiqqandan keyin x, y va z o'zgaruvchilar qiymatlari mavjud bo'lmaydi. Shunday qilib, ushbu o'zgaruvchilar o'z qiymatlarini funksiya parametrlarining qiymatlariga faqat lokal tarzda - funksiya qobig'ining ichidagina o'zgartiradi.

Har qanday funksiya qobig'ida aniqlangan o'zgaruvchi singari sws o'zgaruvchi ham lokal o'zgaruvchidir. Dastlab uning qiymati aniqlanmagan bo'ladi. Funksiyaning ichida u sws=0.5 qiymatni qabul qiladi. Funksiyadan qaytgandan keyin funksiyada qo'llanilganligiga qaramasdan, u noaniq bo'lib qoladi. Agar sws ni chiqarishga harakat qilinsa, komandalar oynasida xatolik to'g'risida axborot hosil bo'ladi. Bunga ishonch hosil qilish uchun bitta misolni ko'raylik.

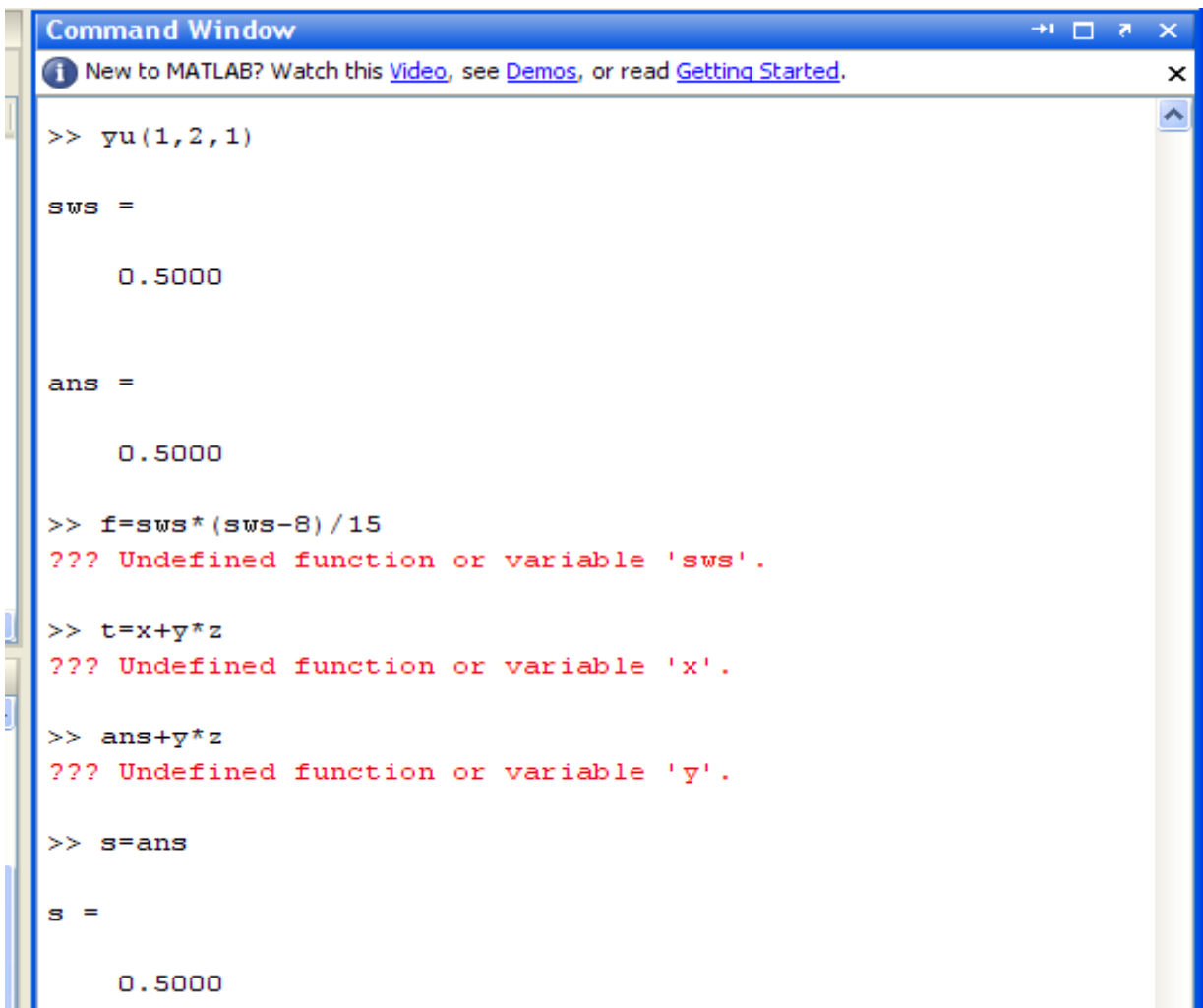
Komandalar oynasida quyidagi hisoblashlarni ko'ramiz:

```
>> yu(1,2,1)
```

```
sws = 0.5
```

```
ans = 0.5
```

```
>>sws??? Undefined function or variable sws'.
```



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> yu(1,2,1)

sws =

    0.5000

ans =

    0.5000

>> f=sws*(sws-8)/15
??? Undefined function or variable 'sws'.

>> t=x+y*z
??? Undefined function or variable 'x'.

>> ans+y*z
??? Undefined function or variable 'y'.

>> s=ans

s =

    0.5000
```

7.9 – rasm. Lokal o'zgaruvchilar buyruqlar oynasida.

Ko'rinib turibdiki, lokal o'zgaruvchilar komandalar oynasida qiymatga ega emas.

Funksiyadagi hamma amallar bajarilgandan keyin, yani fayl-funksiyaning oxiriga yetilgandan keyin funksiyadan qaytiladi. Funksiya qobig'ida shartli operatorlar, sikllar yoki tanlash operatori ishlatilganda funksiyaning ma'lum joyidan qaytish zaruriyati hosil bo'lishi mumkin. Buning uchun return komandasi xizmat qiladi. Har qanday holda ham funksiya chiqish parametrlarining qiymatlarini qaytaradi. Yuqoridagi misolda sws o'zgaruvchisi chiqish parametri bo'lib hisoblanadi.

7.7. O'zgaruvchi sondagi argumentli funksiyalar

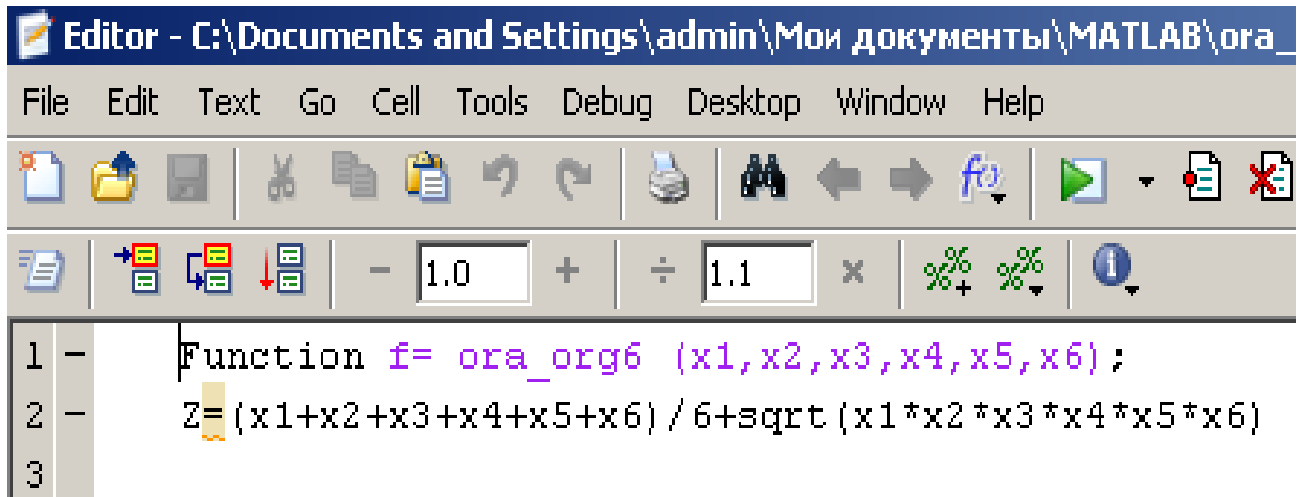
Maxsus xususiyatlarga ega bo'lgan funksiyalarni yaratishda quyidagi ikki funksiya foydali bo'lishi mumkin:

- nargin -berilgan funksiyadagi kirish parametrlarining sonini qaytaradi;

- nargout -berilgan funksiyadagi chiqish parametrlarining sonini qaytaradi.

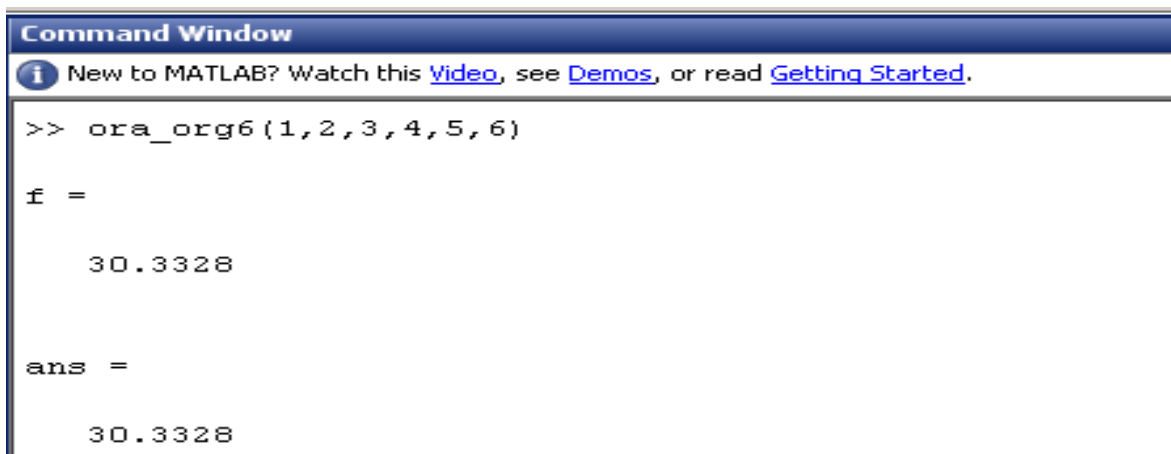
Aytaylik, oltita $x_1, x_2, x_3, x_4, x_5, x_6$ argumentlar o'rta arifmetigi va o'rta geometrigining yig'indisini hisoblaydigan funksiya yaratish zarur bo'lsin.

Odatdagi yo'l bilan ora_org6 nomli funksiyanı yaratib ko'ramiz.



7.10 - rasm. Oddiy fayl-funksiya.

Uning ishlashini nomiga murojaat qilib tekshirib ko'ramiz:



7.11- rasm. Oddiy fayl-funksiyaga murojat.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> ora_org6(1,2,3,4,5,6)

f =

    30.3328

ans =

    30.3328

>> ora_org6(1,2,,4,5,)
??? ora_org6(1,2,,4,5,)
      |
Error: Expression or statement is incorrect--possibly unbalanced (, (, or

>> ora_org6(1,2,4,5,)
??? ora_org6(1,2,4,5,)
      |
Error: Unbalanced or unexpected parenthesis or bracket.

>> ora_org6(1,2,4,5)
??? Input argument "x5" is undefined.

Error in ==> ora_org6 at 4
f=(x1+x2+x3+x4+x5+x6)/6+sqrt(x1*x2*x3*x4*x5*x6)

>> ora_org6(2,4,5)
??? Input argument "x4" is undefined.

Error in ==> ora_org6 at 4

```

7.12 - rasm. Oddiy fayl-funksiyaning ishlashini tekshirish.

Shunday qilib, oltita argument bo'lganda funksiya to'g'ri ishlaydi. Lekin argumentlar soni oltitadan kam bo'lsa, xatolik to'g'risida axborot beradi. Har qanday sondagi kirish argumentlari bo'lganda (yuqoridagi misol uchun oltitagacha) to'q'ri ishlaydigan oraorg6 nomli funksiyani yaratish uchun nargin funksiyasidan foydalanamiz:

```

Editor - C:\Documents and Settings\admin\Мои документы\MATLAB\ora_org6.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons: Save, Copy, Paste, Undo, Redo, Print, Run, etc.]
- 1.0 + ÷ 1.1 x % + % ↓ ⓘ
1
2
3 function f= oraorg6(x1,x2,x3,x4,x5,x6)
4 n=nargin
5 if n==1 f=x1+sqrt(x1) end
6 if n==2 f=(x1+x2)/2+sqrt(x1*x2) end
7 if n==3 f=(x1+x2+x3)/3+sqrt(x1*x2*x3) end
8 if n==4 f=(x1+x2+x3+x4)/4+sqrt(x1*x2*x3*x4) end
9 if n==5 f=(x1+x2+x3+x4+x5)/5+sqrt(x1*x2*x3*x4*x5) end
10 if n==6 f=(x1+x2+x3+x4+x5+x6)/6+sqrt(x1*x2*x3*x4*x5*x6) end
11

```

7.13 - rasm. O'zgaruvchi argumertli fayl-funksiya.

Funksiyani ishlashini tekshirib ko'ramiz:

```

>> oraorg6(1)
ans = 1
>> oraorg6(1,2)
ans = 0.1414
>> oraorg6(1,2,3)
ans = 0.8165
>> oraorg6(1,2,3,4)
ans = 0.5103
>>
>> oraorg6(1,2,3,4,5)
ans = 0.2739
>> oraorg6(1,2,3,4,5,6,7)
??? Error using ==> oraorg6
Too many input arguments.

```

Shunday qilib, kirish parametrlarining soni 1 dan 6 tagacha bo'lganda fayl-funksiya ishlaydi va 6 tadan ko'p bo'lganda hisoblashlar to'g'risida xatolik haqida axborot chiqadi. Bu axborotni interpretatorga birlashtirilgan xatoliklarni diagnostika qilish tizimi beradi. Olingan

axborotga qarab fayl-funksiyadagi xatoliklar to'g'rilanadi yoki yangi funksiya yaratiladi.

Nazorat savollari

1. Ma'lumotlarning qanday turlari mavjud?
2. Fayllar nechta turga bo'linadi?
3. Ishchi fayllar qanday aniqlanadi?
4. Fayl-ssenariyning tuzilishi qanday?
5. Ishchi fayllarning xususiyatlarini ayting.
6. Fayllarga qanday kengaytma beriladi?
7. Ishchi fayllarga qanday nomlar berish mumkin?
8. Fayllarning qanday toifalar mavjud?
9. Ma'lumotlarning qanday toifalarini bilasiz?
10. Fayllarda izohlar qaysi pozitsiyadan boshlanishi kerak?
11. M-fayl funksiya nima?
12. M-fayl funksiya qanday xossalarga ega?
13. Lokal va global o'zgaruvchilarni tushuntirib bering.
14. nargin va nargout qanday funksiyalar?
15. Fayl-funksiyada matnli sharhlarni tushuntiring

Mustaqil ishlash uchun misollar

1. Quyidagi berilgan qatorlarni 2 tadan birlashtiring:
 $a='function'$, $b='off y=f(x)'$, $c='real'$, $d='i2'$
2. 12 va 13 sonlarni MATLAB funksiyasi yordamida qatorlarga aylantiring.
3. 923 va 2409 sonlarini ikkilik sanoq sistemasidagi mos sonlarga aylantiring.
4. 1011101111 va 10101011110110 qator ko'rinishidagi ikkilik sanoq sistemasidagi sonlarni o'nli sanoq sistemasidagi mos sonlarga aylantiring.
5. $ax^2+bx+c=0$ ko'rinishida berilgan kvadrat tenglamaning yechimlarini aniqlovchi fayl-ssenariy tuzing.
6. $y=e^x \cos x$, $x \in [a,b]$ funksiya grafigini chizuvchi fayl-ssenariy tuzing.
7. Uchburchakning tomoni va unga tushirilgan balandligi bo'yicha yuzini va perimetrini topish uchun fayl-ssenariy tuzing.
8. Chiziqli tenglamalar sistemasini Kramer usulida yechish uchun fayl-ssenariy tuzing.

9. $f(x,y)=\sqrt{|3x + 4y|} + (x+y)^2\sin(x+y)$ funksiya qiymatlarini fayl- funksiya yordamida hisoblang.

10. Hisoblang: $y=5\sin\sqrt{3\pi + 7}+\text{tg}(2\pi - 9)$;

11. $y=x^4\cos x$, $x\in[a,b]$ funksiya grafigini chizuvchi fayl-ssenariy tuzing.

12. $f(x,y) = x + xy^2 - 4xy + y^4$ funksiyaning qiymatlarini hisoblovchi fayl- ssenariy tuzing.

13. Kvadrat tenglamani yechish uchun fayl-funksiya yarating. Kirish parametrlari sifatida kvadrat uchrad koeffitsiyentlarini oling.

14. $Z = \frac{x + y + \sin(x + y)}{3} + \cos(x - y)$ funksiya qiymatlarini hisoblash uchun fayl-funksiya yarating.

15. $y = a\sin x + b\cos x$, $x \in [a_1, b_1]$ funksiya grafigini chizuvchi fayl-funksiya tuzing. Kirish parametrlari sifatida a , b , a_1 , b_1 larni oling.

16. $y = \sin x + (1 - x)\cos x$, $x \in [a, b]$ funksiya qiymatlarini hisoblovchi fayl-funksiya yarating. Kirish parametrlari sifatida a , $b \in [0, 1]$ larni oling.

17. Argumentlarining soni 1 dan 8 gacha o'zgaruvchi funksiyaning hisoblash uchun fayl-funksiya yarating.

18. Chiziqli tenglamalar sistemasini iteratsiyalar usuli bilan yechish uchun fayl-funksiya yarating. Kirish parametri sifatida asosiy matritsani va ozod hadni oling.

19. Matlabdagi ($:$) komandasi yordamida 2 ta arifmetik va 2 ta kamayuvchi geometrik progressiya tuzib, ularni n ta hadi yig'indisini hisoblovchi fayl-funksiya va fayl-ssenariy yarating.

20. Chiziqli tenglamalar sistemasini teskari matritsa usuli bilan yechish uchun fayl-ssenariy tuzing.

21. 2 ta matritsani shunday tuzinki, ular ustida qo'shish, ayirish, ko'paytirish amallarini Matlabda bajarish mumkin bo'lsin.

22. 3 ta matritsa tuzib, ular ustida ustunlar va qatorlar bo'yicha 180° ga, soat strelkasiga qarshi 90° ga burish amallarini bajaring.

8. MATLABDA DASTURLASH ASOSLARI. SHARTLI VA SIKL OPERATORLARI

MATLAB tizimida har bir foydalanuvchi uchun dastur tuzish imkoniyati bor. Bu dasturlar keyinchalik alohida funksiya sifatida ishlatilishi mumkin. Dasturlashda hisoblashlarni bajarilishini boshqarish va nazorat qilish maqsadida Matlabda maxsus konstruktsiyalardan foydalaniladi. Bu konstruktsiya (operator)larning har biri alohida yoki ichma-ich joylashgan bo'lishi mumkin. Har bir boshqarish operatori o'ziga mos operatorni yopilishini bildiruvchi end bilan tugagan bo'lishi kerak. Matlabda boshqarish konstruktsiya(operator) lariga while, for, if, va switch-case kabilar kiradi.

8.1. Sikl operatorlari

Matlabda ko'rsatilgan operatorlar ketma-ketligini ma'lum marta takrorlab bajarish uchun for...end sikl operatoridan foydalaniladi. Uning formati quyidagicha:

```
for <sikl hisoblagich> =<x0:h:xn>
    {operatorlar}
end
```

Sikl qobig'ini tashkil qiluvchi operatorlar ketma-ketligi <sikl hisoblagich>ning boshlang'ich qiymat x_0 dan boshlab h qadam bilan oxirgi qiymati x_n gacha bo'lgan qiymatlarida bajariladi. Agar qadam h berilmasa, tizim uni avtomatik tarzda 1 deb hisoblaydi.

Misollar: 1)

```
for i=1:9
    for j=1:10
        ad(i,j)=i^2+j^2-3*(i+j)-1;
    end
end
```



```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> for i=1:9
    for j=1:10
        ad(i,j)=i^2+j^2-3*(i+j)-1;
    end
end
>> ad

ad =

    -5    -5    -3     1     7    15    25    37    51    67
    -5    -5    -3     1     7    15    25    37    51    67
    -3    -3    -1     3     9    17    27    39    53    69
     1     1     3     7    13    21    31    43    57    73
     7     7     9    13    19    27    37    49    63    79
    15    15    17    21    27    35    45    57    71    87
    25    25    27    31    37    45    55    67    81    97
    37    37    39    43    49    57    67    79    93   109
    51    51    53    57    63    71    81    93   107   123

```

8.1 - rasm. Sikl operatorlari.

Bu dastur ishlashi natijasida (9x10) o'lchovli matritsa hosil qilinadi. Komandalar oynasida yuqoridagi operatorlar ketma-ketligini hosil qilib, natijani matritsa ko'rinishida yoki matritsa elementlari ko'rinishida olish mumkin(8.1- rasm.)

```

2) for i=0:2:10
    x(i)=exp(i-5.3)-2;
    y(i)=x(i).*sin(x(i)-1)-1;
end

```

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> for j=1:2:10
    x(j)=exp(j-5.3)-2;
    y(j)=x(j).*sin(x(j)-1)-1;
end
>> y

y =

Columns 1 through 6
    -0.6930         0    -0.5450         0    -0.0276         0

Columns 7 through 9
     1.1509         0   -10.5793

```

8.2 - rasm. Hisoblanmagan elementlarni aniqlash.

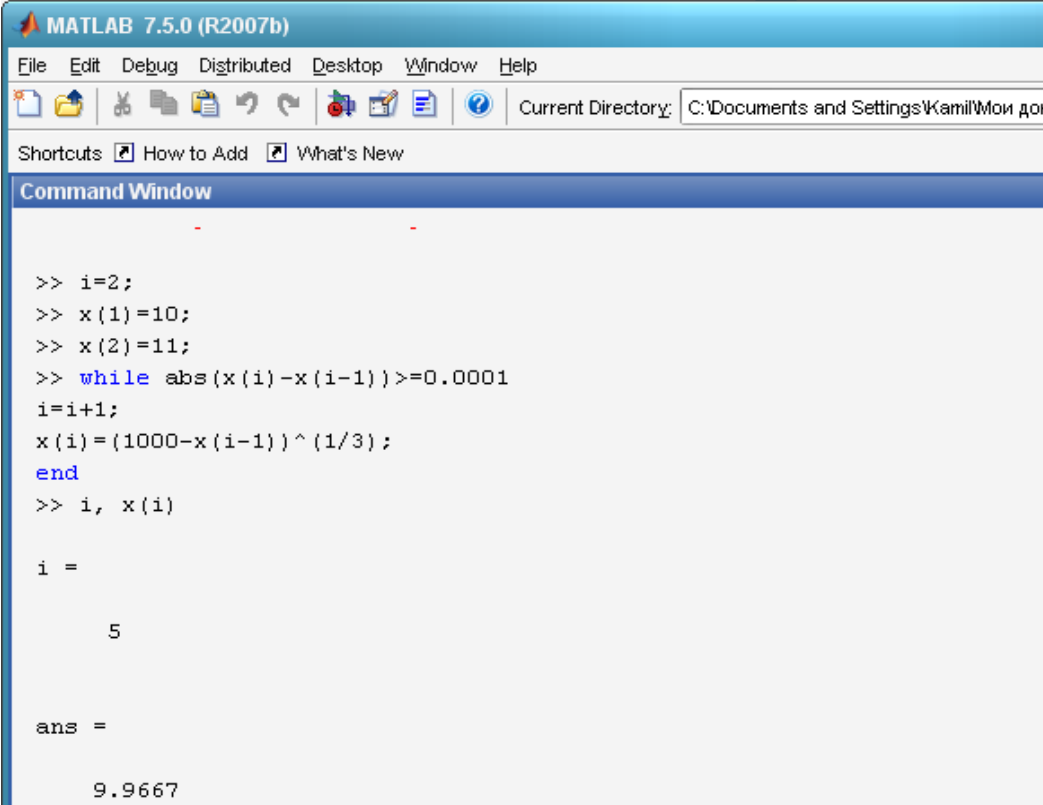
Bu misolda qadam $h=2$ deb olingani uchun y vektorning toq indeksli koordinatalarining qiymatlari berilgan formula bo'yicha hisoblab chiqarilgan, juft indeksli koordinatalarining qiymatlarini esa sistema nol qiymat bilan to'ldirgan.

Matlabda shartli ifodalar bilan ishlaydigan `while...end` ko'rinishidagi sikl operatori ham mavjud bo'lib, uning umumiy ko'rinishi quyidagicha bo'ladi:

```
while <ifoda>
    {operatorlar}
end
```

Bunda {operatorlar} ketma-ketligi <ifoda> “yolq'on” qiymat qabul qilguncha takror bajarilaveradi, <ifoda> xuddi shartli operator if dagi kabi mantiqiy amallar orqali aniqlangan bo'lishi kerak. Masalan, quyidagicha:

```
i=2; x(1)=10; x(2)=11;
while abs(x(i)-x(i-1))>=0.0001
i=i+1;
x(i)=(1000-x(i-1))^(1/3);
end
>>i , x(i)
```



The screenshot shows the MATLAB 7.5.0 (R2007b) Command Window. The window title is "MATLAB 7.5.0 (R2007b)". The menu bar includes "File", "Edit", "Debug", "Distributed", "Desktop", "Window", and "Help". The current directory is "C:\Documents and Settings\Kamil\Мои док". The Command Window contains the following code and output:

```
>> i=2;
>> x(1)=10;
>> x(2)=11;
>> while abs(x(i)-x(i-1))>=0.0001
i=i+1;
x(i)=(1000-x(i-1))^(1/3);
end
>> i, x(i)
```

The output shows:

```
i =
    5

ans =
    9.9667
```

8.3 - rasm. While - end operatoridan foydalanish.

Ketma-ketlikda yozilgan kod (8.3-rasm) $x^3+x=1000$ tenglamaning 0.0001 aniqlikdagi taqribiy yechimini ketma-ket yaqinlashish (iteratsiya) usuli yordamida topib beradi. Bunda, while...end operator qobiq'idagi hisoblashlar necha marta bajarilishi noma'lum. Hosil qilingan z vektorning birinchi komponentasi hisoblashlar sonini bildirsa, ikkinchi komponentasi $x(i)$ esa yechimni bildiradi.

8.2. Tayinlash va shartli operatorlar

Matlabda dasturlash komandalar rejimida va m-fayllarda amalga oshiriladi. Shuni ta'kidlash lozimki, dasturlash m-fayllarda osonroq tuziladi, chunki unda ixtiyoriy qatordagi xatoliklarni to'g'rilash imkoniyati mavjuddir. Bu tizim shunday tuzilganki, komandalar rejimida hisoblash uchun ishlatiladigan o'zgaruvchilarni qiymati berilmagan bo'lsa, ular ustida har qanday amalni bajarish mumkin bo'lmay qoladi. Tayinlash operatori sifatida o'zgaruvchilarga qiymat berish komandasi bo'lgan oddiy "=" tenglik belgisi ishlatiladi. Demak, tayinlash operatori qiymat o'zlashtiruvchi har bir o'zgaruvchi va funksiyalarning qiymatlarini aniqlashda ishlatiladi.

Shartli o'tish operatori if ning formatlari bilan tanishib chiqamiz. Umumiy holda if operatorining formati:

```
if <1-shart>
    { operator1}
elseif <2-shart>
    { operator2}
else
    { operator3}
end
```

ko'rinishida bo'ladi. Agar 1-shart "rost" bo'lsa, boshqarish {operator1}ni bajarishga uzatiladi. Aks holda, yani 1-shart "yolg'on" bo'lsa, u holda boshqarish 2-shartni tekshirishga uzatiladi. Agar y "rost" bo'lsa, boshqarish {operator2}ni bajarishga uzatiladi, aks holda boshqarish {operator3}ni bajarishga uzatiladi.

Yuqoridagi formatda shartlar sifatida mantiqiy va solishtirish amallari yordamida boq'langan algebraik ifodalar ishlatilishi mumkin. Masalan,

```
for i=1:6
    for j=1:6
        if i==j
```

```

        a(i,j)=i+j+2;
elseif abs(i-j)==1
        a(i,j)=-1;
else
        a(i,j)=1;
end
end
end
>>a

```

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

THE INPUT CHARACTER IS NOT VALID IN MATLAB STATEMENTS
OR EXPRESSIONS.

>> for i=1:6
    for j=1:6
        if i==j
            a(i,j)=i+j+2;
        elseif abs(i-j)==1
            a(i,j)=-1;
        else
            a(i,j)=1;
        end
    end
end
end
a
a =

     4     -1      1      1      1      1
    -1      6     -1      1      1      1
     1     -1      8     -1      1      1
     1      1     -1     10     -1      1
     1      1      1     -1     12     -1

```

8.4 - rasm. Shartli va sikl operatorlari.

Komandalar ketma-ketligi (6x6) o'lchovli matritsani hosil qiladi (8.4-rasm.).

Shartli operatorning qisqa formatlaridan ham foydalanish mumkin:

- a) if <shart>
 - {operatorlar}
 - end
- b) if <shart>

```
{operatorlar1}  
else  
{operatorlar2}  
end
```

8.3. Tanlash operatori

Dasturni bajarish yo'lini ko'rsatib beruvchi vositalardan biri tanlov operatori `switch` hisoblanadi. Uning formati quyidagicha bo'ladi:

```
Switch <tekshiriluvchi ifoda >  
case <qiymat >  
    operator, operator,...;  
    case { 1- qiymat, 2- qiymat,... }  
        operator, operator,...;  
otherwise,  
    operator, operator,... ;  
end
```

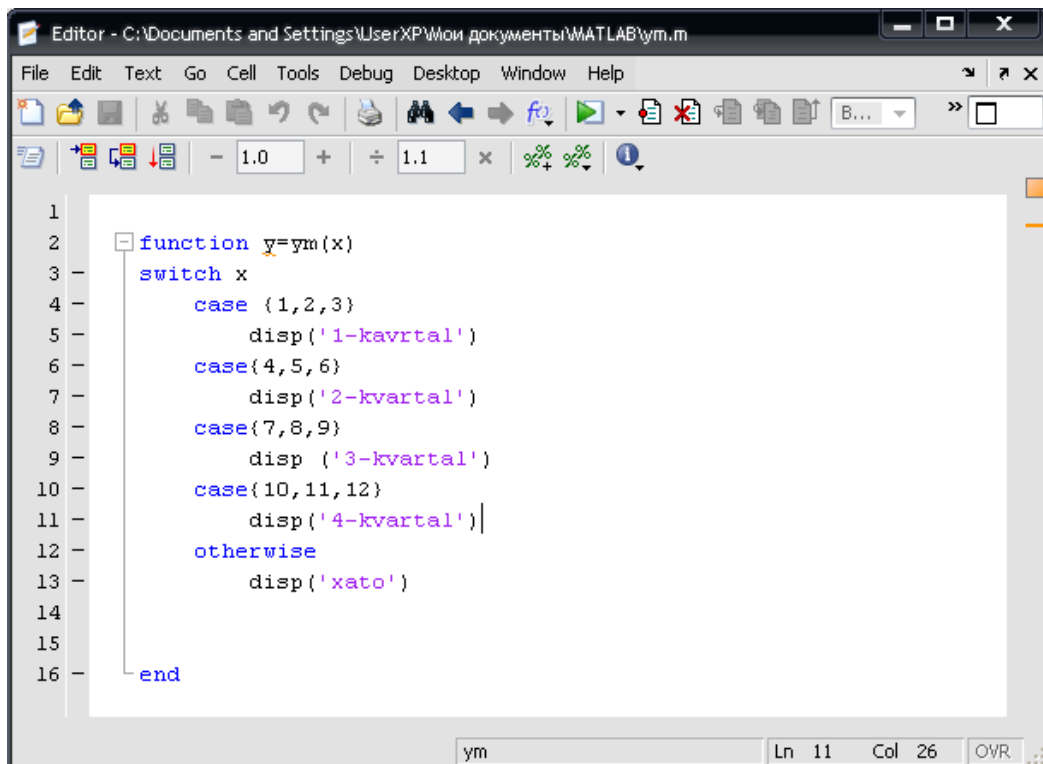
Bu operatorlar formatidagi <tekshiriluvchi ifoda>-skalyar ifoda yoki simvolli qator bo'lishi mumkin. Operator quyidagicha ishlaydi: <tekshiriluvchi ifoda> case ostidagi < qiymat>ga teng bo'lsa, u holda ko'rsatilgan operatorlar bajariladi, aks holda otherwise dan keyingi operatorlar bajariladi. Simvolli qator bo'lgan holda agar `strcmp(<tekshiriluvchi ifoda>,<qiymat>)` "rost"ni bersa, <tekshiriluvchi ifoda>ning <qiymatga> tengligi "rost"ni beradi. Tanlov operatorini qo'llashga doir misollar ko'ramiz.

1) Faraz qilaylik, method o'zgaruvchisi mavjud va simvolli bo'lsin.

U holda switch operatorini quyidagicha ishlatiladi:

```
switch lower (method)  
case { 'chiziqli', 'bichiziqli' }, disp { ' chiziqli usul' }  
case { 'cubic' }, disp ('cubic usul')  
case { 'nearest' }, disp ('taqribiy usul')  
otherwise, disp ('noma'lum usul')  
end
```

2) `ym.m` nomli `m`-fayl yaratamiz:



8.5 - rasm. Tanlov operatorining qo'llanishi.

va quyidagicha natijani olamiz:

```

>> ym(1)
1-kavrtal
>> ym(4)
2-kvartal
>> ym(8)
3-kvartal
>> ym(12)
4-kvartal
>> ym(15)
xato

```

3) x vektorning yiq'indisi a sonidan oshmaydigan, birinchisidan boshlab ketma-ket kelgan barcha koordinatalari aniqlansin. Bu masalani hal qiluvchi komandalar ketma-ketligi quyidagicha bo'ladi:

```

>>x,a; k=0;      s=0;
while s<=a
k=k+1;  s=s+x(k);
    if s>a
        break
    else y(k)=x(k); end
end

```

```

function y=f(x,a)
k=0;    s=0;
    while s<=a
        k=k+1;    s=s+x(k);
        if s>a
            break
        else    y(k)=x(k);
        end
    end
end

```

8.6 - rasm. Yangi vektor hosil qilish.

Fayl- funksiyaga murojaat qilib natijalar olish mumkin bo'ladi:

```
>>x=1:10;
```

```
>>y=f(x,6)
```

```
    y=1 2 3
```

```
>>y=f(x,11)
```

```
    y=1 2 3 4
```

4) Yuqoridagi 3-misolni if...end operatori yordamida bajarish ham mumkin.

8.4. Hisoblashlarda pauzalar hosil qilish

Dasturning ishlashini vaqtincha to'xtatib turish uchun pause operatoridan foydalaniladi. U quyidagi shakllarda ishlatilishi mumkin:

- pause – hisoblashlar biror klavisha bosilguncha to'xtab turadi;
- pause(N) - hisoblashlar N sekundga to'xtaydi;
- pause on - pause ni qayta ishlash rejimini ulaydi;
- pause off - pauze ni qayta ishlash rejimini uzadi;

Quyidagi pause.m deb nomlangan m-faylni ko'raylik:

```
x=0:0.1:10; pause
```

```
y=sin(x); plot(x,y)
```

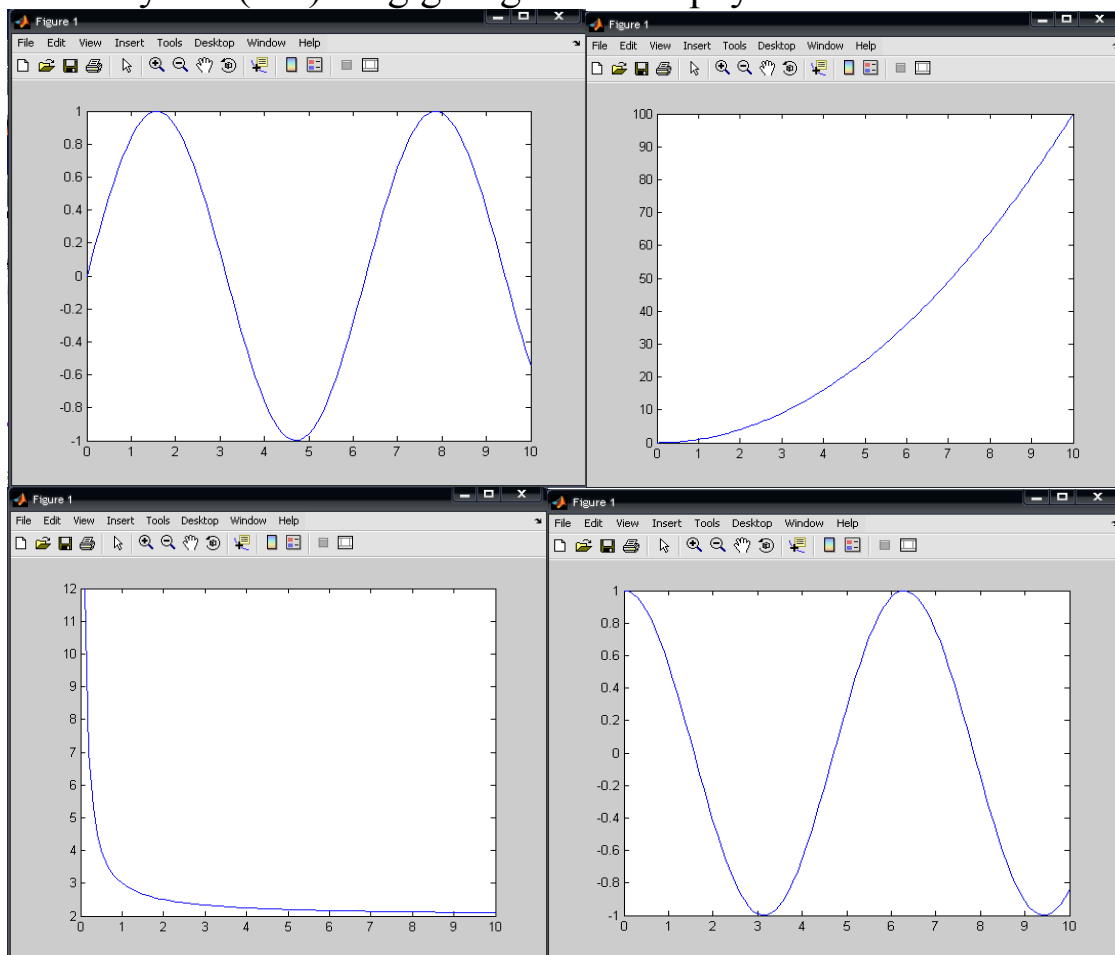
```
y1=cos(x) ; pause(12)
```

```
plot(x,y1) , pause(15)
```

```
y2=x.^2; plot(x,y2) pause(20)
```

$y_3 = 1./x+2$; plot(x,y3)

Ushbu dastur F5 klavishi yoki komandalar oynasidan pauze komandasi yordamida ishga tushirilgandan keyin pauze operatori ta`sirida biror klavisha bosilguncha kutib turadi. Klavisha bosilgandan keyin sin(x) ning grafigi quriladi. Keyingi grafiklar pause(N) operatorlarning ishlashiga asosan ma`lum vaqt oraliqlaridan keyin ketma-ket quriladi, yani 12 sekunddan keyin cos(x) ning, 15 sekunddan keyin x^2 ning va 20 sekunddan keyin $1/(x-2)$ ning grafigi ekranda paydo bo`ladi.



8.7 - rasm. Pauzalar bilan hosil qilingan grafiklar.

Nazorat savollari

1. Matlabda boshqaruvchi strukturalar deganda nima tushuniladi?
2. Shartli operator konstruksiyasini tushuntirib bering.
3. Sikl operatorlari konstruksiyalarini tushuntirib bering
4. Tanlash operator konstruksiyasini tushuntirib bering.
5. Hisoblashlarda to`xtashlar qanday hosil qilinadi?

9. DASTURNI SOZLASH

9.1. Dasturni sozlash komandalari

Dasturni sozlash - dasturni tayyorlash jarayoni kabi muhimdir. Shuning uchun quyida bu jarayonni alohida qadamlarga bo'lib qarab chiqamiz. Matlab sistemasida m-fayllarni sozlashning asosiy vositasi - bu sozlangan zamonaviy grafik interfeysli **muharrir/sozlagich**(M-file editor/debugger)dir, lekin Matlab komandalar rejimida ham sozlashning asosiy imkoniyatlarini beradi.

m-fayllardan komandalar rejimida sozlashga o'tish uchun **“keyboard”** komandasini berish kerak . Uni komandalar rejimida ham qo'llash mumkin :

```
>> keyboard
K>> type sw1
switch var
case {1,2,3}
disp('birinchi kvartal')
case {4,5,6}
disp('ikkinchi kvartal')
case {7,8,9}
disp('uchinchi kvartal')
case {10,11,12}
disp('to'rtinchi kvartal')
otherwise
disp('beshimchi kvartal')
end
K>> return
```

Sozlash rejimiga o'tishning belgisi sifatida **“k>>”** ko'rinadi. Bu belgi **“return”** komandasidan keyin **“>>”** belgiga qaytadi. Xuddi shu jarayon **“dbquit”** komandasidan keyin ham bajariladi, faqat bunda m-faylning bajarilishi ham tugallanadi(sozlash jarayoni bilan birgalikda).

Agar **“return”** komandasi m-faylning ichida bo'lsa, u faylning bajarilishini to'xtatadi va boshqaruvni fayl chaqirilgan joyga beradi.

9.2. m-fayl listingi satrlarini raqamlab chiqarish

m-fayllarni sozlashning usullaridan biri - bu unda uzilish nuqtalarini joylashtirishdir. Bunday nuqtalarda dastur bajarilishi to'xtaydi va

dasturning analizini, masalan o'zgaruvchilar qiymatlarini ko'rishni boshlash mumkin bo'ladi, ammo komandalar rejimida bunday nuqtalarni o'rnatish "sichqoncha" orqali mumkin emas. Shuning uchun satrlarni dastur listingiga raqamlab chiqarish kerak. Bu "**dbtype**" komandasi yordamida amalga oshiriladi.

```
>> keyboard
K>> dbtype sw1
1  switch var
2  case {1,2,3}
3  disp('birinchi kvartal')
4  case {4,5,6}
5  disp('ikkinchi kvartal')
6  case {7,8,9}
7  disp('uchinchi kvartal')
8  case {10,11,12}
9  disp('to'rtinchi kvartal')
10 otherwise
11 disp('beshinchi kvartal')
12 end
```

9.3. Uzilish nuqtalarini o'rnatish , olib tashlash va ko'rib chiqish

Tekshirilayotgan m-fayllarda uzilish nuqtalarini o'rnatish uchun quyidagi komandalar ishlatiladi:

- dbstop in M-file at lineno-berilgan satrda uzilish nuqtasini o'rnatish.
- dbstop in M-file at subfun-ost funksiyalarda uzilish nuqtasini o'rnatish.
- dbstop in M-file- m-faylda uzilish nuqtasini o'rnatish.
- dbstop if error-xatolik haqida axborotda uzilish nuqtasini o'rnatish, faqat "**try...catch**" sikli ichidagi xatoliklardan tashqari.
- dbstop if all error —ixtiyoriy xatolik haqidagi axborotda uzilish nuqtasini o'rnatish.
- dbstop if warning-ogoxlantirish haqidagi axborotda uzilish nuqtasini o'rnatish.
- dbstop if infnan yoki naninf - "inf" yoki "NaN" axboroti chiqqanda uzilish nuqtasini o'rnatish.

Bu komandalarni "in", "at" va "if" so'zlarisiz ham ishlatish mumkin .
Masalan :

- `dbclear` M-file at line no- berilgan faylning berilgan qatoridan uzilish nuqtasini o'rnatish.

Joriy sessiyadan o'rnatilgan uzilish nuqtalari ro'yxatini chiqarish uchun “`dbstatus`” komandasi ishlatiladi. Masalan ,

```
K>> dbstatus
```

```
Breakpoint for S:\MATLAB\bin\demo1.m is on line 2.
```

```
Breakpoint for S:\MATLAB\bin\sd.m is on line 3.
```

9.4. m-faylni bajarilishini boshqarish

Uzilish nuqtalarini o'rnatilgandan keyin m-faylni tekshirish jarayonini boshlash mumkin. Qadamba- qadam tekshirish uchun “`dbstep`” komandasi quyidagi formatlarda ishlatiladi:

- `dbstep`- navbatdagi qadamning bajarilishi.

- `dbstep nlines`- dasturning ko'rsatilgan sondagi satrlarining bajarilishi.

- `dbstep in`- agar joriy m-faylning navbatdagi bajarilayotgan satri boshqa m-fayldan chaqirilayotgan funksiya bo'lsa, bu format chaqirilayotgan funksiyaning birinchi bajarilayotgan satriga o'tishga va shu yerda to'xtashga imkon beradi.

- `dbstep out`- agar joriy m-faylning navbatdagi bajarilayotgan satri m-fayldan chaqirilayotgan funksiya bo'lsa, bu format chaqirilayotgan joyga o'tishga va u bajarilgandan keyin darhol to'xtashga imkon beradi.

Dasturning bitta to'xtalishidan ikkinchisiga o'tish uchun “`dbson`” komandasi ishlatiladi.

9.5. Ishchi fazoni ko'rish

Uzilish nuqtalarida ishchi sohani “`who`” va “`whos`” komandalari orqali ko'rish mumkin. Bundan tashqari ishchi sohada chaqirilgan funksiyalarni yuqoriga va pastga harakatlantirish uchun quyidagi komandalar ishlatiladi:

- `dbdown`-yuqoridan pastga

- `dbup`-pastdan yuqoriga

- Funksiyalarning harakatini ko'rish uchun “`dbtack`” komandasi ishlatiladi.

- Sozlashni tugallash uchun “`dbquit`” komandasi ishlatiladi.

9.6. m-fayllarni profillash

•Dasturni sozlash bu - dasturning ishlash protsedurasini amalga oshirish garovidir. Shu bilan birgalikda dasturni bajarilish vaqtini minimallashtirish yoki kodlar hajmini minimallashtirish, yani dasturni optimallashtirish masalasi ham juda muhimdir.

•Dasturning alohida qismlarini bajarilish vaqtini baholash - uni profillash deyiladi.

•Bu protsedurani bajarish uchun “profile” komandasi ishlatiladi. U quyidagi qator opsiyalarga ega :

INFO = profile- quyidagi maydonlar bilan strukturani qaytaradi:

- file-profillanayotgan ochiq yo'l .
- interval-vaqt intervali(sekundlarda).
- count-o'lchovlar vektori .
- state-profillovchining holati:
- “on”(ulangan) yoki “off”(uzilgan)

Ta'kidlash joizki, Matlab profillash vositalari faqat m-fayl funksiyalarini tahlil qilishga imkon beradi. Ssenariy fayllarini profillash uchun ularni fayl -funksiyaga o'tkazish kerak.

m-faylni profillashga misollar:

1. Yakobi elliptik funksiyasi - “ellipj”

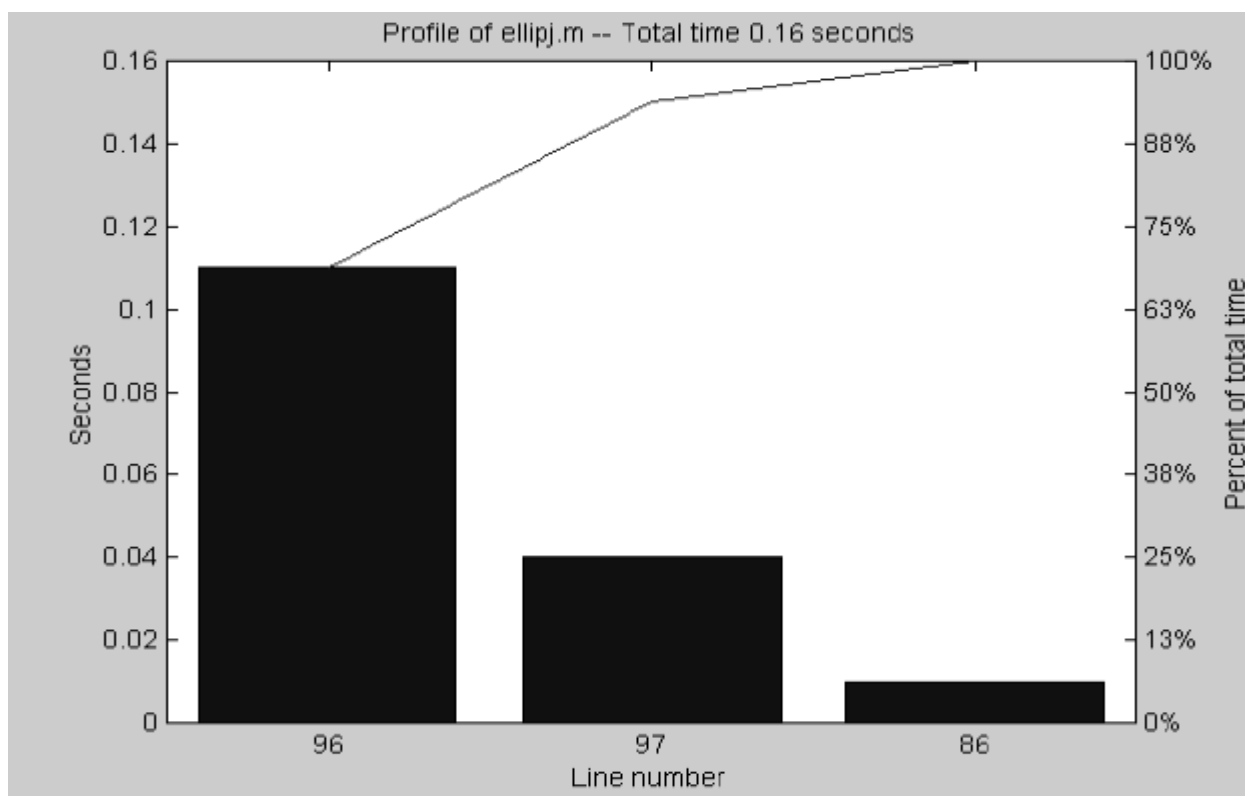
```
>> profile on
>> profile ellipj
>> ellipj([0:0.01:1],0.5);
>> profile report
```

Total time in "S:\MATLAB\toolbox\Matlab\specfun\ellipj.m": 0.16 seconds 100% of the total time was spent on lines:

```
[96 97 86]
85: if ~isempty(in)
0.01s, 6% 86: phin(i,in) = 0.5 * ...
87:(asin(c(i+1,in).*sin(rem(phin(i+1,in),2*pi)))./a(i+1,in))
95: m1 = find(m==1);
0.11s, 69% 96: sn(m1) = tanh(u(m1));
0.04s, 25% 97: cn(m1) = sech(u(m1));
98: dn(m1) = sech(u(m1));
>> INFO=profile
```

```
INFO = file: 'S:\MATLAB\toolbox\Matlab\specfun\ellipj.m'
interval: 0.0100
count: [98x1 double]
```

```
state: 'off'  
>> profile plot
```



9.1-rasm. Profillash natijalarining grafik tasvirlanishi

Nazorat savollari

1. Dasturni sozlash deganda nimani tushunasiz ?
2. Sozlash komandalaridan bir nechtasini keltiring
3. m-fayl listingi satrlari qanday raqamlanadi ?
4. Uzilish nuqtalari nima uchun kerak ?
5. Uzilish nuqtalari qanday o'rnatiladi va qanday olib tashlanadi ?
6. Ishchi sohani ko'rish qanday amalga oshiriladi ?
7. Profillash deganda nimani tushunasiz ?

10. XATOLIKLARNI QAYTA ISHLASH

Dastur foydalanuvchi uchun zarur bo'lgan harakatlarni bajarmasa, bunday dastur xato dastur hisoblanadi. Matlab tizimida xatoliklar diagnostikasi katta ahamiyatga ega. Kiritilayotgan buyruq va ifodalarni tekshiradi va xatolar to'g'risida axborot yoki ogohlantirish beradi. Ular turli sabablarga ko'ra dasturda uchraydi.

10.1. Xatoliklar haqidagi axborot

Aksariyat hollarda hisoblash jarayonida xatoliklar yuzaga keladi. Masalan, $\sin(x)/x$ funksiya hisoblanganda $x=0$ bo'lgan holatda “nolga bo'lish” degan xabar chiqadi. Xatolikning yuzaga kelishi bilan, xatolik haqidagi xabar chiqishi bilanoq hisob to'xtatiladi. Shuni aytib o'tish kerakki, har qanday xato hisoblashlarni to'xtatishiga olib kelavermaydi. Matlabda “xatolik haqida ogohlantirish” (Warning so'zidan keyin) va “xatolik haqida axborot” (??? belgidan keyin) farqlanadi. “Ogohlantirish”da hisoblashlar to'xtamaydi, “Xatolik haqida axborot”dan keyin esa hisoblashlar to'xtaydi.

Quyidagi tur xatoliklarni sanab o'tish mumkin:

- **Sintaksis xatoliklar:**

Matlab tizimida mavjud bo'lmagan o'zgaruvchini aniqlashga murojat qilinsa, masalan, `hsin(1)`, tizim xatolik haqida quyidagi axborotni chiqaradi:

```
>> hsin(1)
??? Undefined function or variable 'hsin'
```

Bu misolda giperbolik sinusni hisoblaydigan funksiyaning nomi noto'g'ri yozilgani uchun tizim `hsin` nomli funksiya yoki o'zgaruvchi ichki funksiyalar ichida ham, `m`-funksiyalar ichida ham aniqlanmaganini ko'rsatayapti. Agar nom to'g'ri kiritilsa, hisoblash amalga oshadi:

```
>> sinh(1)
ans =
    1.1752
```

- **Hisoblashlardagi xatoliklar:**

```
>> 1/0
ans =
    Inf
```

yani nolga bo'lish natijasida mashina cheksizligining qiymatini anglatuvchi tizim o'zgaruvchisini chiqaradi.

```
>> 0/0
```

```
ans =
```

```
NaN
```

Yani aniqmaslik natijasida ma'lumotlarning sonli xarakterga ega emasligini ko'rsatuvchi tizim o'zgaruvchisini chiqaradi.

• **Matritsalar ustida arifmetik amallar bajarilgandagi xatoliklar:** ularning o'lchamlari mos kelmaganda, masalan, matritsalar ko'paytirilganda ularning o'lchamlari mos kelmasa, quyidagicha xatolik haqida axborot beradi:

```
>> w=[1 2 3;4 5 6; 7 8 9]
```

```
w = 1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
>> k=[1 2;4 5]
```

```
k = 1 2
```

```
4 5
```

```
>> n=w*k
```

```
??? Error using ==> mtimes
```

```
Inner matrix dimensions must agree
```

Yani ko'paytirish amali noto'g'ri ishlatilgan, matritsalarining o'lchovi mos bo'lishi kerak.

O'zgaruvchilar kompyuterning ishchi soha deb ataluvchi ma'lum joyini egallaydi. Ishchi sohani tozalashda clear funksiyasidan foydalanilganda aniqlanishlari o'chirilgan o'zgaruvchi noaniq bo'lib qoladi va keyinchalik undan foydalanishga harakat qilinsa, xato to'q'risida axborot chiqaradi. Masalan ,

```
>> x=2*pi
```

```
x =
```

```
6.2832
```

```
>> v=[1 2 3 4 5]
```

```
v =
```

```
1 2 3 4 5
```

```
>> clear x
```

```
>> x
```

```
??? Undefined function or variable 'x'.
```

Shuning uchun, clear komandasidan ehtiyot bo'lib foydalanish zarur.

10.2. Xatoliklarni bildiruvchi error va warning komandalari

- **Error komandasidan foydalanish**

Xatolik to'q'risidagi axborotni chiqarish uchun error ('Xatolik to'q'risidagi axborot') komandasi xizmat qiladi. Xatolik to'q'risidagi axborotni beruvchi komanda kirgizilgan $sd(x)=\sin(x)/x$ funksiyaning hisoblanish dasturini ko'raylik:

```
function f=sd(x)
if x==0 error('Xatolik – nolga bo'lish')
end
f=sin(x)/x
```

Natijasi quyidagicha bo'ladi:

```
» sd(1)
f =0.8415
ans =0.8415
» sd(0)
??? Error using ==> sd
Xatolik – nolga bo'lish
```

- **Warning komandasidan foydalanish**

Agar xatolik yuz berganda ham hisoblashlar davom etishi kerak bo'lsa, warning ('Ogohlantiruvchi axborot') komandasidan foydalanish mumkin:

```
function f=sd(x)
if x==0 warning('Ogohlantiruvchi axborot')
end
f=sin(x)/x
```

Natijasi quyidagicha:

```
» sd(1)
f =
    0.8415
ans =
    0.8415
```

```
» sd(0)
```

Warning: Ogohlantiruvchi axborot

10.3. Lasterr funksiyasi va xatoliklarni qayta ishlash

Tajribali dasturchilar xato yuzaga kelish vaziyatini nazarda tutishlari kerak. Masalan, yuqoridagi misolda $x=0$ da $\sin(x)/x=0/0=1$ deb olish va shu hisob uchun 1 qiymatdan foydalanish to'q'ri bo'ladi:

```
function f=sd0(x)
if x==0 f=1; else f=sin(x)/x; end
return
```

Bu holatda x ning turli qiymatida natija aniq chiqadi:

```
>> sd0(1)
ans =
    0.8415
>> sd0(0)
ans =     1
```

Lasterr funksiyasi so'nggi bo'lib o'tgan xato haqidagi xabarni chiqarish uchun foydalaniladi. Masalan:

```
>> aaa
??? Undefined function or variable 'aaa'.
>> 2+3
ans =
     5
>> 1/0
ans =
    Inf
>> lasterr
ans =
    Undefined function or variable 'aaa'.
```

Lasterr funksiyasi ??? belgidan keyin keluvchi matnli xabarni qaytaradi.

10.4 . varargin va varargout o'zgaruvchilari

Quyida aniqlanadigan “**varargin**” va “**varargout**” o'zgaruvchilari funksiyalarda o'zgaruvchi sondagi kirish va chiqish parametrlaridan foydalanishga imkon beradi:

1. varargout=foo(n)–foo funksiyaning o'zgaruvchi sondagi chiqish parametrlari ro'yxatini qaytaradi;
2. y=function bar (varargin) – bar funksiyaga o'zgaruvchi sondagi argumentlarni beradi.

“**varargin**” va “**varargout**” o’zgaruvchilari funksiyalarning ixtiyoriy sondagi argumentlarini faqat m – file funksiyalar qobig’ida aniqlaydi.

Funksiya argumentlarini yozishni soddalashtirish uchun ularni yacheykalar massivi bo’lgan maxsus o’zgaruvchi **varargin** orqali aniqlanadigan ro’yxat kabi ifodalash mumkin. U kichik xarflar bilan yozilishi kerak va u o’z ichiga argumentlarni, shuningdek, funksiya opsiyalarini olishi mumkin. Masalan:

```
function myplot(x,varargin)
plot(x,varargin{:})
function [s,varargout] = mysize(x)
nout = max(nargout,1)-1;
s = size(x);
for l=1:nout, varargout(i) = {s(i)};
end
```

Bu o’zgaruvchi o’ziga barcha kiruvchi parametrlarni va ikkinchi argument boshlanuvchi opsiyalarini oladi. Ushbu funksiyaga quyidagicha

```
myplot(sin(0:.1:1),’color’.[.5 .7 .3],’linestyle’,’:’)
```

murojat qilinganda **varargin** 1x4 o’lchamli massiv yacheykalarini ifodalaydi, u o’ziga quyidagi qiymatlarni oladi:

```
‘color’, [.5 .7 .3], ‘linestyle’ u ‘:’.
```

varargin singari **varargout** o’zgaruvchisi ham turli sondagi chiquvchi parametrlarni massiv yacheykalariga birlashtiradi. Bu o’zgaruvchi **varargin** kabi argumentlar ro’yxatida so’ngida bo’lishi shart. Bu o’zgaruvchi odatda funksiya chaqirilayotganda vujudga kelmaydi. Quyida sikl yordamida keltirilgan misolni ko’rib chiqaylik:

```
function [s,varargout] = mysize(x)
nout = max(nargout,1)-1;
s = size(x);
for i=1:nout.
varargout(i)= {s(i)}; end
```

Ushbu misolda sikl yordamida **varargout** o’zgaruvchisining ikkinchi qiymatidan boshlab barcha parametrlari birlashtiriladi.

10.5. M-fayl funksiyalarni bajarilish xususiyatlari va izohlar haqida

Matlab murakkab hisoblar uchun ishlatilishi sababli ularning tavsiflari yaqqol va tushunarli bo’lishi kerak. Buning uchun izohlar qo’llaniladi.

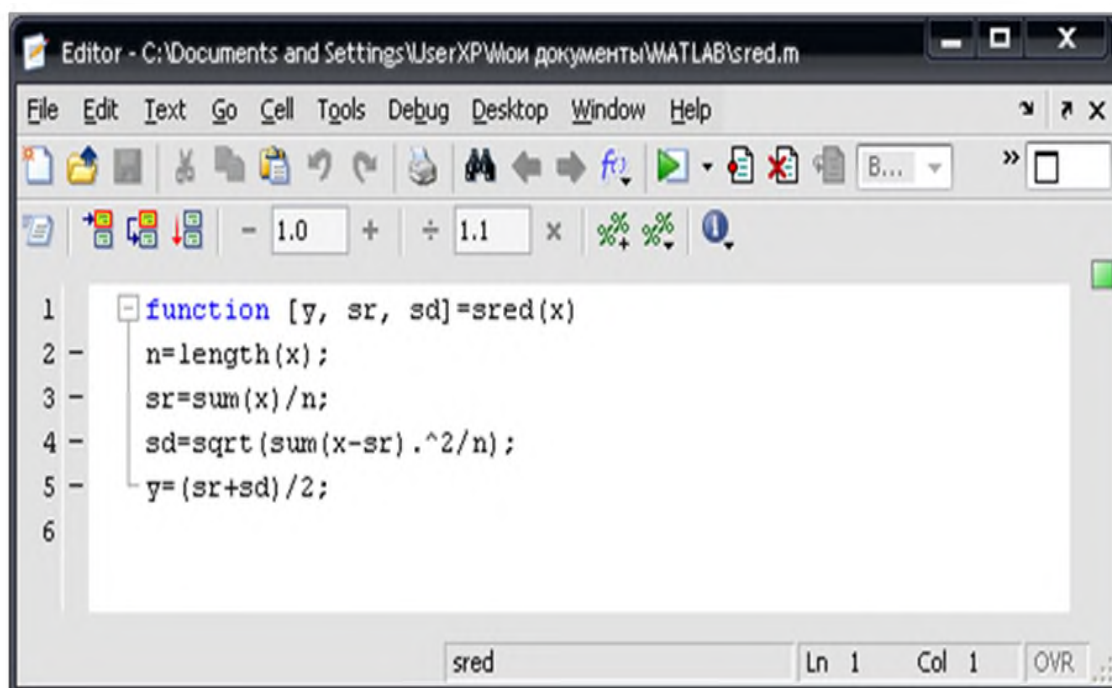
Izohlar % simvoli yordamida kiritiladi, masalan: $Z=X+Y$ %Z massivi X va Y massivlarining yig'indisi.

Odatda m-fayllarning birinchi satrlari help «Fayl_nomi» buyruq'idan keyin ekranga chiqariluvchi, ular to'q'risidagi qisqacha axborot bo'ladi. Yetarli darajada mukammal izohlarning m-fayllarga kiritilishi keyinchalik ular bilan ishlashni osonlashtiradi.

help catalog komandasi, catalog - m-faylli katalog nomi, barcha kataloglar uchun umumiy bo'lgan izohlarni chiqaradi. Foydalanuvchi m-fayl redaktori yordamida mustaqil yaratishi mumkin bo'lgan bunday izohlar katalogning contents.m. faylida saqlanadi. Agar bunday fayl bo'lmasa, barcha m-fayl kataloglari uchun izohlarning birinchi qatorlari ro'yxatini chiqaradi.

Umumiy ko'rinishda m-fayldagi funksiya quyidagicha bo'ladi:

Function $y=<\Phi$ ункция_номи>



```
Editor - C:\Documents and Settings\UserXP\Мои документы\MATLAB\sred.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 function [y, sr, sd]=sred(x)
2     n=length(x);
3     sr=sum(x)/n;
4     sd=sqrt(sum(x-sr).^2/n);
5     y=(sr+sd)/2;
6
```

10.1-rasm. Izohsiz fayl-funksiya.

m-fayl-funksiyalar komandalar rejimida ishlatilishi mumkin, shuningdek boshqa m-fayllardan chaqirilishi mumkin. Bu holda barcha kirish va chiqish parametrlarini ko'rsatish zarur. Global o'zgaruvchilar qo'llanilganda ular barcha berilgan masalani yechishda ishlatiladigan m-fayllarda va ular tarkibiga kiruvchi ichki funksiyalarda ham e'lon qilinishi zarur. Funksiya nomlari yagona bo'lishi kerak. MATLAB tizimi har bir yangi nom paydo bo'lganda bu nom o'zgaruvchilarga tegishlimi, ushbu m-

fayldagi ichki funksiyami yoki PRIVATE katalogiga tegishli funksiya ekanligini tekshiradi.

Katta hajmdagi ma'lumotlarga ega bo'lgan masalalarni yechishda operativ xotiraning yetishmasligi seziladi. Buning belgisi sifatida «Out of memory» xabarining chiqishidir. Bu holatda quyidagi choralarni ko'rish foydali:

- katta hajmdagi keraksiz eski ma'lumotlarni o'chirish;
- foydalanilayotgan ma'lumotlarning hajmini kamaytirish;
- foydalanilayotgan xotira hajmini cheklashni bekor qilish;
- kompyuterning fizik xotira hajmini oshirish.

Kompyuterning operativ xotira qurilmasi hajmi qancha katta bo'lsa, xatolarning yuzaga kelishi shuncha kam bo'ladi.

10.6. P-kodlarni yaratish

P-kodlar (psevdokod) m-fayl ko'rinishidagi ssenariylar yoki funksiyalarni sintaktik nazorat qilish bilan boq'liq, bu esa hisobni biroz sekinlashtiradi. Vaqtinchalik p-kodlar xotirada clear komandasi ishga tushguncha yoki ish seansi tugaguncha saqlanadi. Bundan tashqari, Matlab p-kod ssenariylari va funksiyalarini pcode komandasi yordamida tashkil qilish va saqlash mumkin. Masalan: pcode M-fayl_nomi pcode *.m

Bu komandaning qo'llanilishi asosan murakkab deskriptor grafikada va GUI vositalarini yaratishda foydalidir. Bu holda hisoblashlarni tezligi sezilarli darajada oshadi. Agar foydalanuvchi ishlab chiqqan m-fayllarini va undagi amalga oshirilgan g'oyalarni va algoritmlarni yashirishni istasa, p-kodlar ular uchun foydalidir.

Quyidagi misolni ko'rib chiqaylik:

```
told=cputime;  
x=-15:.0001:15;  
plot(x,sin(x))  
t=cputime-told
```

Yuqorida keltirilgan dastur nuqtalarning katta miqdori bo'yicha sin(x) funksiyaning grafigini quradi. Shuningdek, u berilgan ssenariyning bajarilish vaqtini sekundlarda hisoblaydi. Ishga tushirganda quyidagilarni olamiz:

```
» rr  
t=  
0.4400
```

Endi p-kodlarni yaratishni bajaramiz va yana dasturni ishga tushiramiz:

```
» pcode rr
```

```
» rr
```

```
t=
```

```
0.3900
```

```
» rr
```

```
t =
```

```
0.3300
```

Bu natijalardan hisoblash vaqti qanchalik tezlashgani ko'rinib turibdi.

Nazorat savollari

1. Xatolik haqida axborot nima?
2. Ogohlantirish qanday axborot?
3. Qayta ishlash tushunchasi nima?
4. Lasterr funksiyasining vazifasi nima?
5. varargin va varargout nima?
6. Izohlar qanday ifodalanadi?
7. P-kodlarni yaratish mexanizmlari qanday?
8. Qachon P-kodlarni yaratish maqsadga muvofiq?

11. OB`YEKTGA MO`LJALLANGAN DASTURLASH

11.1. Ob`yektning sinfini tekshirish

Biz Matlab tizimini o`rganishda har xil ob`yektlarni ko`p marta ishlatdik, lekin ularga alohida ob`yekt sifatida ahamiyat bermadik. Masalan, figure ob`yekti, ishlatiladigan har xil sonlar, vektorlar, matritsalar va h.k. Bular esa ob`yektga mo`ljallangan dasturlashning belgilaridan hisoblanadi va bu belgi tashqi belgidir.

Ob`yektga mo`ljallangan dasturlashning asosini uchta holat belgilaydi: **Inkapsulatsiya** – ma`lumotlarni va dasturlarni birlashtirish va ularni funksiyalarning kiruvchi va chiquvchi parametrlari orqali uzatish. Dasturlashni bunday elementi ob`yekt deyiladi. Bu dasturni qandaydir monolit, bo`linmas narsa sifatida olib qaramay, ko`plab mustaqil elementlarga bo`lish imkonini beradi. Har bir element alohida modul sifatida olib qaraladi. Inkapsulatsiya tarjimasigermetik berkitilgan, tashqi ta`sirlardan himoyalangan dastur qismi deganidir.

-Me`rosxo`rlik (nasledovanie) - yangi ob`yektlarni tuzish va ularning xossalarni o`zida saqlab qolgan tegishli (docherniy) ob`ektlarni hosil qilish. Bir necha ob`ektlarni xossalarni saqlab qoluvchi ob`ektlar sinfini ham hosil qilish mumkin. Me`rosxo`rlikka ma`lumotlarning turlarini berish va boshqa dasturlash elementlari kiradi. Me`rosxo`rlik yordamida paydo bo`lgan ob`yekt metod va xususiyatlari 3 ta ko`rinishga ega bo`lishi mumkin:

1) o`rniga qo`yish(almashtirish) - yangi ob`ekt ajdodlarining xususiyatlarini shunchaki o`zlashtirib olmaydi, balki unga ta`rif ham beradi;

2) yangi sinf yoki ob`yekt butunlay yangi metodlar yoki xususiyatlarni qo`shadi;

3) rekursiv, yangi ob`yekt o`z ajdodlarini xususiyatlarini to`q`ridan-to`q`ri olib qoladi.

-Polimorfizm - yuqoridan pastgacha hosil qilingan ob`yektlar ketma-ketligida ishlatiluvchi qandaydir harakatga bir xil nom berish. Bu shunday holatki, bunda qandaydir bitta sinf ko`p shakllarga ega bo`ladi. Dasturlashda ko`p shakllar deganda bitta nom bilan avtomatik mexanizm tomonidan tanlab olingan turli kodlarning nomidan ish qilish tushuniladi. Polimorfizm yordamida bitta nom turli xususiyatlarni bildirishi mumkin.

Bulardan tashqari Matlabning o`zida obyektlar qismlarini birlashtirish va bir nechta ob`ektlarni birlashtirish imkoniyati mavjud.

Ob'ektni aniq bir sinfga tegishli qandaydir struktura kabi aniqlash mumkin. Matlabda obyektlni yettita asosiy sinfi mavjud:

- double -ikkilangan aniqlikdagi sonli elementlar massivi;
- sparse -ikki o'lchovli sonli va kompleks matritsalar;
- struct -strukturalar (yozuvlar) massivi;
- cell -yacheykalar massivi;
- javaarray -java massivi;
- function_handle -funksiyalar deskriptorlari;
- char- simvollar.

11.1. Obyektning sinfini tekshirish

Biz ba'zi sinflar obyektlni bilan tanishganmiz, lekin ularni qaysi sinfga tegishli ekanligiga urg'u berilmagan. Matlabga xos xususiyatlardan biri shundaki, ob'ektlarning hech qanday sinflari e'lon qilinmaydi (u yangi tuzilgan bo'lsa ham), masalan **name='nom'** o'zgaruvchisini hosil qilib, simvollar massiviga tegishli bo'lgan **name** ob'ektlni olamiz. Bu char sinfga tegishli bo'ladi. Demak, har bir o'zgaruvchi qabul qilgan qiymatiga qarab u yoki bu sinfga tegishli ekanligi aniqlanadi.

O'zgaruvchi ob'ektlnigini aniqlash uchun **isobject(x)** funksiyasi ishlatiladi. Agar x Matlab ob'ekti bo'lsa, isobject(x) funksiyasi 1 natijani beradi, aks holda 0 ni beradi. Ob'ektni va ob'yektlar sinfini hosil qilish uchun **class(x)** operatori ishlatiladi. Bu operator x ob'yektlning sinfini chiqarib beradi (masalan, double, sparse, char, cell va hokazo bo'lishi mumkin).

Ushbu **isa(x, 'name class')** komandasi agar x opostrof ichidagi sinfga tegishli bo'lsa, mantiqiy 1 ni hosil qiladi, aks holda 0 ni beradi. Masalan,

```
>> x=[1 2 3]; isa(x,'char')
ans =0
>> isa(x,'double')
ans =1
```

```
>> x=magic(4)

x =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

>> class(x)

ans =

double

>> class('x')

ans =

char

>> isobject(x)

ans =

    0

>> isa(x,'char')

ans =

    0

>> isa(x,'double')

ans =

    1
```

11.1 - rasm. Ob'yektlarning sinfini aniqlash.

11.2. Handle va inline funksiyalar

Matlabda **handle funksiya** deb ataluvchi alohida ob'yektlar yaratish mumkin. handle funksiyani qurish uchun birlik simvol @ dan foydalaniladi. Masalan, fhsin nomli sinusni qiymatini hisoblovchi handle funksiyasi quyidagicha bo'ladi:

```
>> fhsin=@sin
```

```
fhsin = @sin
```

Bu oddiy funksiya emasligi quyidagidan ko'rinadi:


```
>> fhsin(1)
ans = @sin
```

Ko'rinib turibdiki, bunda hisoblash bajarilmadi, balki handle funksiyaning oddiy aniqlanishi berildi. Demak, handle funksiya o'z nomi bilan xarakterlanadi, lekin argumentga ega emasdir. Bu funksiyaning nomi xuddi fayl- funksiyaning nomi kabi bo'lishi kerak. handle funksiyani hisoblash uchun quyidagi komanda ishlatiladi:

```
feval(<handle funksiya nomi>,< handle funksiya argumentlari>)
```

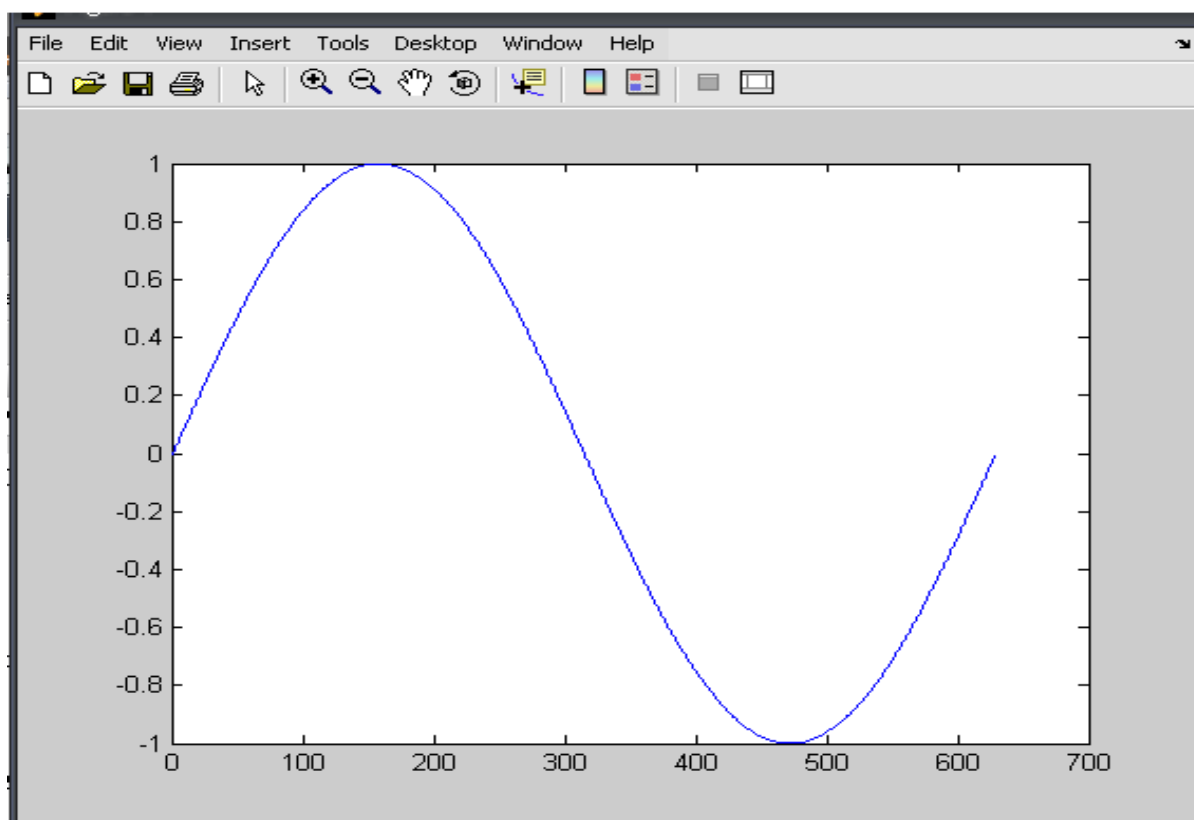
,
bu erda handle funksiya nomi @-belgisiz ishlatiladi.

Endi biz yuqorida hosil qilingan sinusni qiymatini hisoblovchi handle fuksiyani hisoblashimiz mumkin:

```
>> feval(fhsin,1)
ans = 0.8415
```

handle funksiyaning grafigini chizish mumkin, masalan :

```
>> plot(feval(fhsin,0:.01:2*pi))
komandasi yordamida quyidagi grafik chiziladi:
```



11.2-rasm. handle funksiyaning grafigi.

Matlabda foydalanuvchining funksiyalarini beruvchi yana bir muhim funksiyalar sinfi bu **inline funksiyalar**dir. Bu funksiyaning quyidagi ko'rinishlari bor:

```
d=inline ('ifoda') ;  
d=inline ('ifoda', <argumentlar>) ;  
d=inline ('ifoda', <parametrlar >),
```

<parametrlar> quyidagicha p1, p2,... ko'rinishida bo'ladi. Eng muhimi 'ifoda' ixtiyoriy matematik ifoda bo'lishi mumkin, argumentlar esa bitta yoki bir nechta bo'lishi mumkin.

Masalan, $f(x,y) = \sin^2(2x + y) + \cos^2(x - y)$;

```
>> fc=inline('sin(2*x+y)^2+cos(x-y)^2')
```

```
fc = Inline function:
```

```
fc(x,y) = sin(2*x+y)^2+cos(x-y)^2
```

```
>> fc(0,0)
```

```
ans = 1
```

```
>> fc(3,4)
```

```
ans = 0.5879
```

```
>> fc(pi/7,pi/11)
```

```
ans = 1.8307
```



```
Command Window  
New to MATLAB? Watch this Video, see Demos, or read Getting Started.  
>> fc=inline('sin(2*x+y)^2+cos(x-y)^2')  
  
fc =  
  
Inline function:  
fc(x,y) = sin(2*x+y)^2+cos(x-y)^2  
  
>> fc(0,0)  
  
ans =  
  
1  
  
>> fc(3,4)  
  
ans =  
  
0.5879  
  
>> fc(pi/7,pi/11)  
  
ans =  
  
1.8307
```

11.3 - rasm. Foydalanuvchining inline funksiyasi.

Nazorat savollari

1. Ob'yektga mo'ljallangan dasturlashning asosini nechta holat belgilaydi?
2. Polimorfizm nima?
3. Matlabda ob'yektlar sinfini sanab bering.
4. Ob'yektni va ob'ektlar sinfini hosil qilish uchun qanday operator ishlatiladi?
5. isobject(x) funksiyasi vazifasi nima?
6. handle va inline funksiyasi qanday funksiyalar?

Mustaqil ishlash uchun misollar

1. (10x11) o'lchovli A matritsa hosil qiling va uning qator hamda ustunlaridan tuzilgan massivlarning grafigini chizing, deskriptorlarini toping.
2. $X_i = i \cdot h$, $h = 0.2$, $y_i = x_i - (i \cdot h)^2 + i \cdot h - 3$, $i = 1, 15$ $M_i(x_i; y_i)$ nuqtalardan o'tuvchi chiziq grafigini line ob'yektidan foydalanib hosil qiling.
3. $X = (1, 4, 9, 3, -1)$, $y = (1, 7, 4, 'c')$, $z = ('c', 2, 3, 5)$, $t = [x; x.^2; x.^{(1/3)}]$ massivlarining sinfini Matlab komandalari yordamida aniqlang.
4. $F(x, y) = e^{\sin(x+y)} + 5 \cos(x+y) + 6 \cdot 5^{2x+y}$ funksiya qiymatlarini hisoblovchi inline funksiya tuzing.
5. $Y = \cos(3x + 1.8\pi)$ funksiya qiymatlarini hisoblash uchun handle funksiya tuzing.
6. $Y = \sin x + \cos x$ funksiya grafigini $[-2\pi; 2\pi]$ oraliqda handle funksiyadan foydalanib chizing.

12. MATLABDA GRAFIK VA GISTOGRAMMALAR

Matlab tizimining eng katta xususiyatlaridan biri unda grafik chizish imkoniyatining mavjudligidir. Biz Matlabda ikki vektor grafigini chizishning eng sodda va umumiy komandalari bilan tanishamiz. Bu yerda shuni ta'kidlash lozimki, Matlabda vektor deganda koordinatalari bo'yicha aniqlangan oddiy algebraik vektorni ham tushunish mumkin, yoki o'zgaruvchining ketma-ket hosil qilingan qiymatlaridan iborat vektorni ham tushunish mumkin.

Matlabda grafiklarni har xil koordinata sistemalarda qurish mumkin. Bulardan to'q'ri burchakli dekart koordinatalari sistemasi, polyar koordinatalari, sferik va silindrik sistemalarni keltirish mumkin. Bundan tashqari, koordinatalarni bir sistemadagi ko'rinishidan boshqa ko'rinishga o'tkazish mumkin.

12.1. Matlabda oddiy grafik

Dekart koordinatalar sistemasida grafik chizish uchun umumiy bo'lgan ba'zi komandalarni keltiramiz:

- **plot(x,y)**- x va y vektorlar bo'yicha $y=y(x)$ funksiyaning dekart tekisligidagi grafigini hosil qiladi;
- **plot(y)**- y ning y -vektor elementlari nomerlariga nisbatan grafigini yasaydi;
- **plot(x1,y1 ,,x2,u2,...)** - grafik oynada bir nechta chiziqlarni hosil qiladi;
- **semilogx(x,y)**- "x" o'qi logarifmik masshtabda(asos 10ga teng) olinib, y funksiya grafigi chiziladi;
- **semilogy(x,y)** - "y" o'qi logarifmik masshtabda(asos 10 ga teng) olinib, y funksiya grafigi chiziladi;
- **loglog(x,y)** -"x"va "y" o'qlari logarifmik masshtabda olinib, grafik yasaladi;
- **grid on**- koordinatalar sistemasida to'rni hosil qiladi;
- **title ('matn')** - grafik tepasiga matn yozadi;
- **xlabel ('matn')** - "matn"ni "x" o'qi ostiga yozadi;
- **ylabel ('matn')** - "matn"ni " y " o'qi chap tomoniga yozadi;
- **text(x,y,'matn')** - "matn"ni (x, y) nuqtadan boshlab yozadi;
- **bar(x)** – x matritsaning gistogrammasini yasaydi;

➤ **bar(x,y)** – y vektor(matritsa)elementlarining gistogrammasini x vektorning elementlari(ular o'sish tartibida joylashgan bo'lishi kerak)ga mos ravishda joylashtirib chizadi ;

➤ **bar(x,y,width)** yoki **bar(x,width)**- avvalgilarga o'xshash,faqat ustunlarning maxsuslashtirilgan kengligi bilan(avtomatik ravishda width=0.8)) ;

➤ **subplot(m,n,p)**- grafik oynani mxn ta oynachaga bo'ladi (m-gorizontal bo'yicha, n- vertikal bo'yicha bo'linishlar soni), r- oynacha nomeri(satrlar bo'ylab sanalib boriladi);

➤ **plot (x,y,s)** - plot (x,y) komandaga o'xshash, faqat chiziq turini s qatorli konstanta orqali berish imkoniyati ham bor. s konstantaning qiymatlari grafik chiziqning rangini, markerni va turini bildiruvchi quyidagi simvollar bo'lishi mumkin:

Ranglar:

➤	y -----sariq	m-----siyoxrang
➤	c -----xavorang	r -----qizil
➤	g -----yashil	b -----to'q ko'k
➤	w -----oq	k -----qora

Nuqta turi:

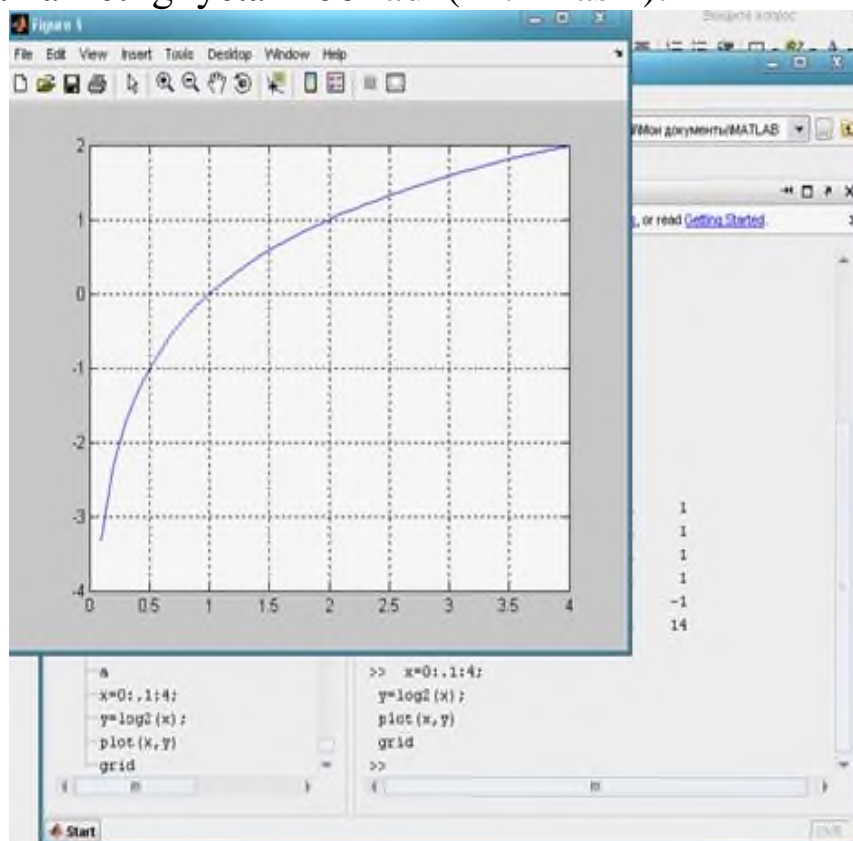
➤	0 -----aylana	. -----nuqta
➤	x -----krest	v -----uchburchak
(pastga)		
➤	+-----plyus	^ -----
uchburchak(yuqoriga)		
➤	*-----yulduzcha	< -----
uchburchak(chapga)		
➤		> -----uchburchak
(o'ngga)		
➤	s -----kvadrat	p -----beshburchak
➤	d -----romb	h -----oltiburchak

Chiziq turi :

➤	- -----uzluksiz chiziq
➤	: -----ikkilangan punktir
➤	-. -----shtrix-punktir
➤	-- -----shtrixli

Bu kattaliklar apostrof ichida ixtiyoriy ketma-ketlikda berilishi mumkin. Dekart koordinatalar sistemasida grafik chizish (x, y) juftligining qiymatlarini aniqlab, hosil bo'lgan nuqtalarni kesmachalar bilan tutashtirish orqali hosil qilinadi. Demak, (x, y) juftliklar soni qanchalik

ko'p bo'lsa, grafik ham shunchalik silliq va aniqroq bo'ladi. Juftliklar avvaldan berilgan bo'lishi yoki ma'lum funksiyaning argumenti va qiymatlaridan hisoblab hosil qilinishi mumkin. Masalan, $y=\log_2(x)$ funksiyaning $x \in [0,4]$ dagi grafigini chizish kerak bo'lsa, quyidagi komandalar ketma-ketligi yetarli bo'ladi (12.1 -rasm):



12.1- rasm. Oddiy grafik.

`plot(x,y)` komandasi grafik oynani ochadi va unda (x,y) juftliklar hosil qilgan grafikni chizadi. Yangi komandani e'lon qilish uchun kursorni komandalar oynasiga o'tkazishimiz kerak. Qayta chizmaslik uchun ... (uch nuqta -qatorni davomi) belgisini ishlatish mumkin:

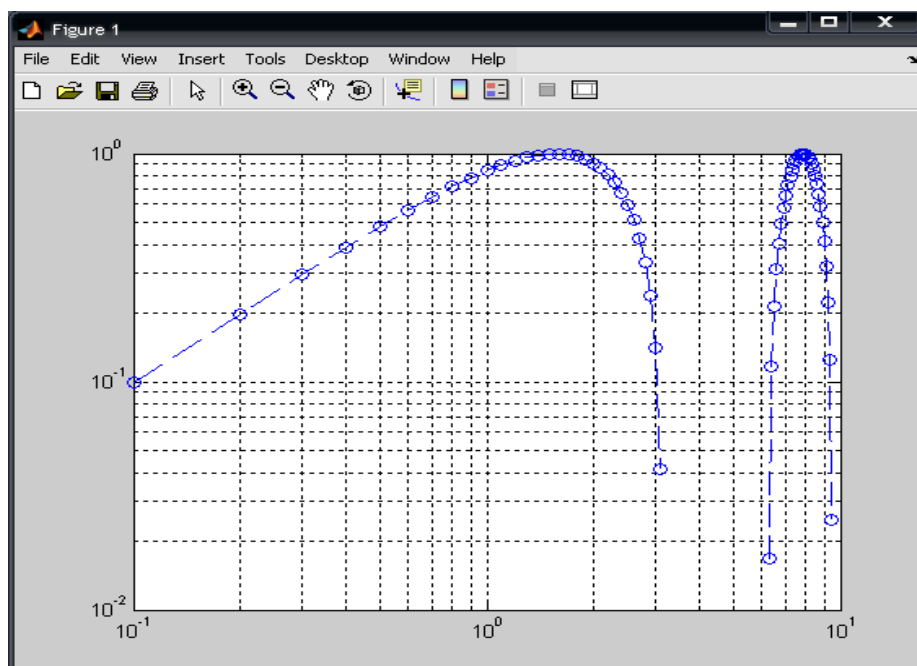
```
>> plot(x,y)...
>> grid,...
>> title('ko'rsatkichli funksiya'),...
>> xlabel('x'),...
>> ylabel('exp(x)'),...
```

Ko'pincha grafik komandalar M-faylga joylashtiriladi (ssenariy- fayl yoki fayl -funksiyalar). Bu usul xatoliklarni to'g'rilash uchun yaxshi imkoniyat beradi. Masalan, fayl-ssenariy sifatida yaratilib , m-faylda gjl nom bilan saqlangan kodni ko'raylik:

```
x=0:1:10;
loglog(x, absos(x),'--ob');
```

grid on

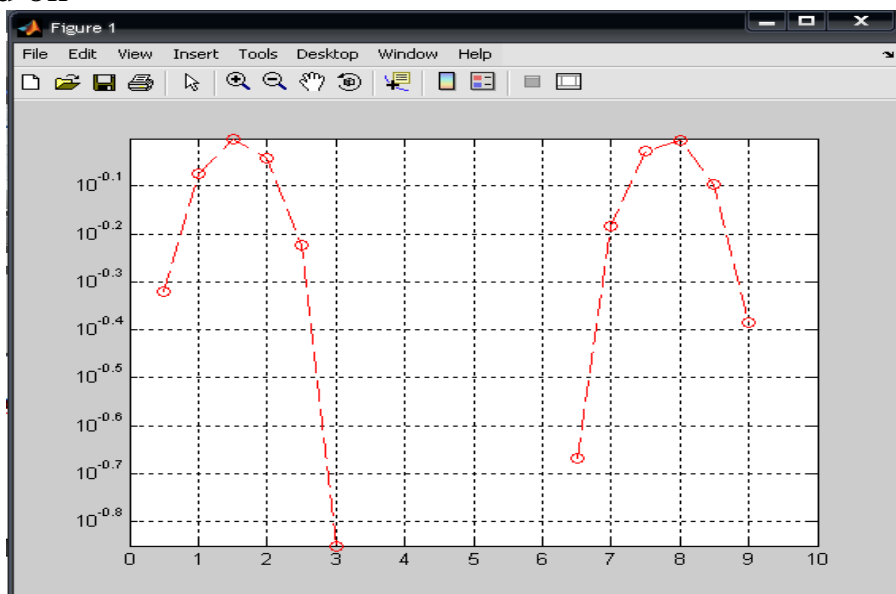
Agar komandalar oynasidan >>gjl deb murojaat etsak, quyidagi natijani olamiz:



12.2-rasm. Uzilishli grafik.

Bu yerda “--“- chiziq turi, “o”- tugun nuqtalar markeri, “b”- chiziqning rangi (blue-havorang). Misol.

```
>> x=0:0.5:10;  
>> semilogy(x,sin(x),'--or')  
>> grid on
```

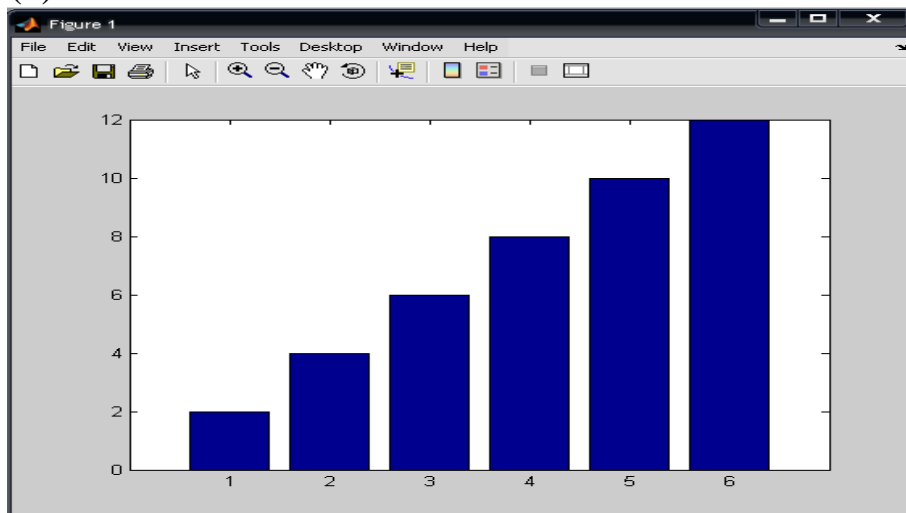


12.3 - rasm. Konturli grafik.

12.2. Gistogrammalar

Amaliy hisoblashlarda biror vektor tarkibini tasvirlaydigan ustunli diagrammalar deb ataluvchi gistogrammalar ko'p uchraydi. Bunda vektorning har bir elementi balandligi uning qiymatiga mos bo'lgan ustun shaklida ko'rsatiladi. Ustunlar tartib raqamlariga va eng baland ustunning maksimal qiymatiga nisbatan ma'lum masshtabga ega bo'ladi. Bunday grafiklar **bar(a)** komandasi yordamida quriladi:

```
>> a=[2 4 6 8 10 12];  
>> bar(a)
```



12.4-rasm. Oddiy gistogramma.

Bundan tashqari gistogramma qurishning yana boshqa usuli ham mavjud bo'lib, bu **hist** funksiyasi yordamida amalga oshiriladi:

- **N=hist(Y)**- avtomatik tanlangan 10 intervalli vektor qiymatini qaytaradi;

- **N=hist(Y,M)**-huddi yuqoridagi kabi, faqat M (M-skalyar) intervalda qaytaradi;

Quyidagi misolni ko'ramiz (12.5-rasm.):

```
>> x=-3:0.2:3; y=randn(1000,1);
```

```
>> hist(y,x); h=hist(y,x)
```

h =

Columns 1 through 13

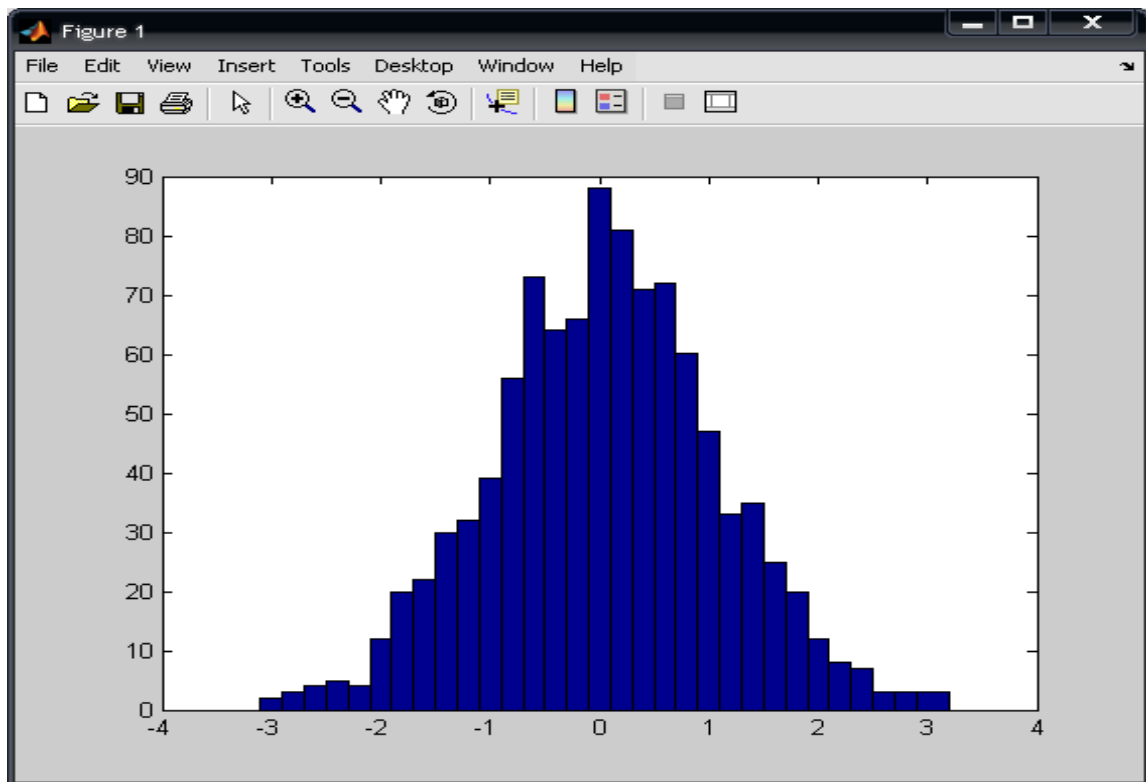
2 3 4 5 4 12 20 22 30 32 39 56 73

Columns 14 through 26

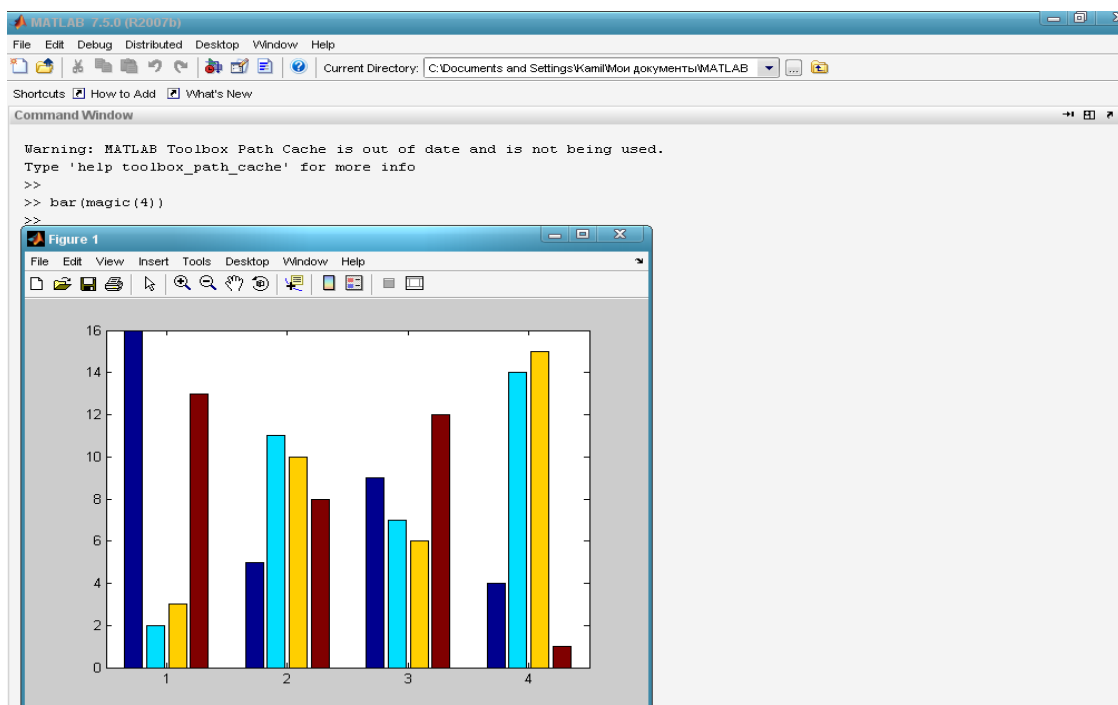
64 66 88 81 71 72 60 47 33 35 25 20 12

Columns 27 through 31

8 7 3 3 3



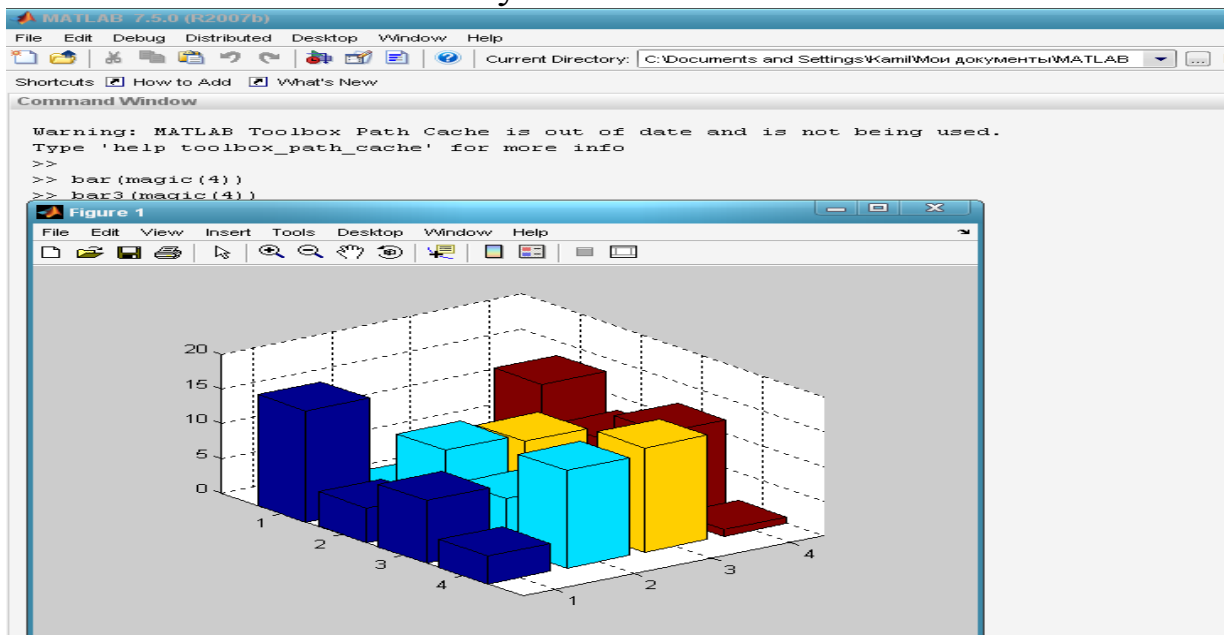
12.5 - rasm. Hist yordamida qurilgan gistogramma.



12.6 - rasm. Matritsaning gistogrammasi.

Yuqoridagi 12.6-rasmda bar komandasini matritsaga qo'llanishi ko'rsatilgan. Unda ketma-ket kelgan 1, 2, 3, 4 raqamlari matritsaning qatorlarini bildiradi. Bunday gistogramma-grafikni uch o'lchovli fazoda

bar3 komandasi yordamida chizish mumkin:



12.7- rasm. Uch o'lchovli gistogramma.

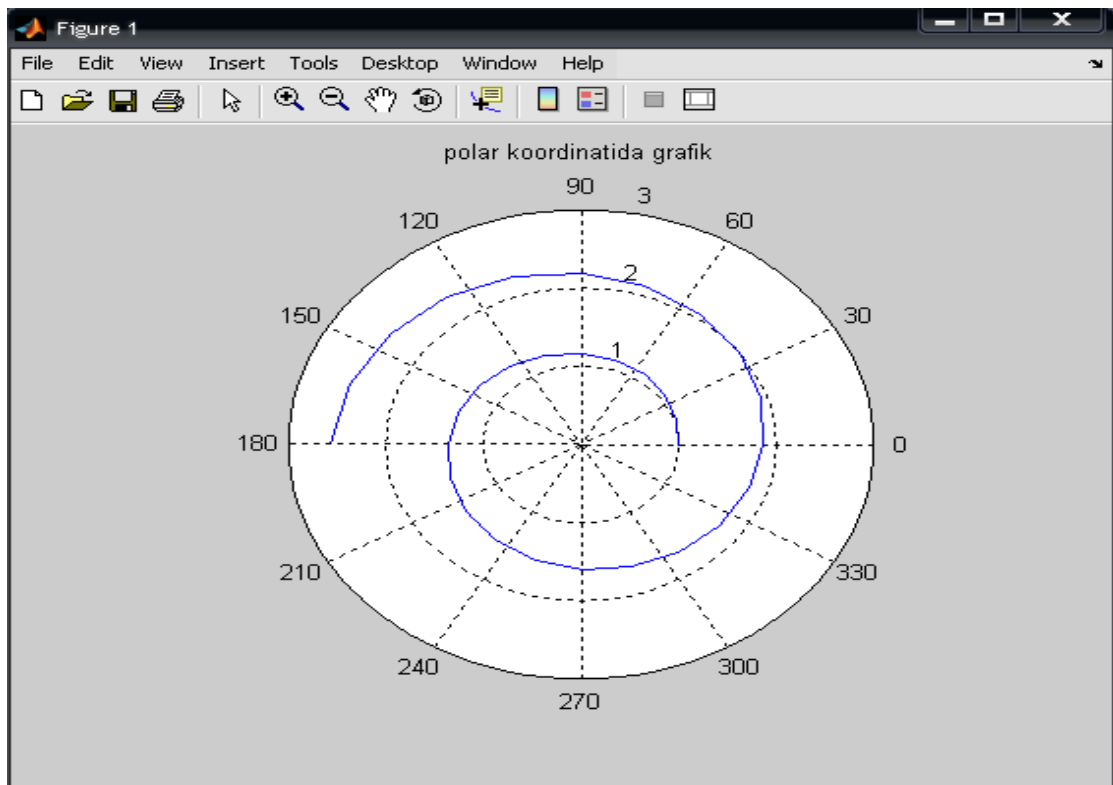
12.3. Polyar koordinatalarda grafik

Polyar koordinatalar tizimida ixtiyoriy nuqta xuddi radius vektor oxiri kabi boshlang'ich koordinatlar tizimidan chiqib, RHO uzunlikka va THETA burchakka egaligini ko'rsatadi. RHO(THETA) funksiya grafigini qurish uchun quyida keltirilgan buyruqlardan foydalaniladi. THETA burchag odatda 0 dan $2 \cdot \pi$ gacha o'zgaradi. Polyar koordinatalar tizimida funksiya grafigini qurish uchun **polyar(...)** tipidagi buyruqdan foydalaniladi :

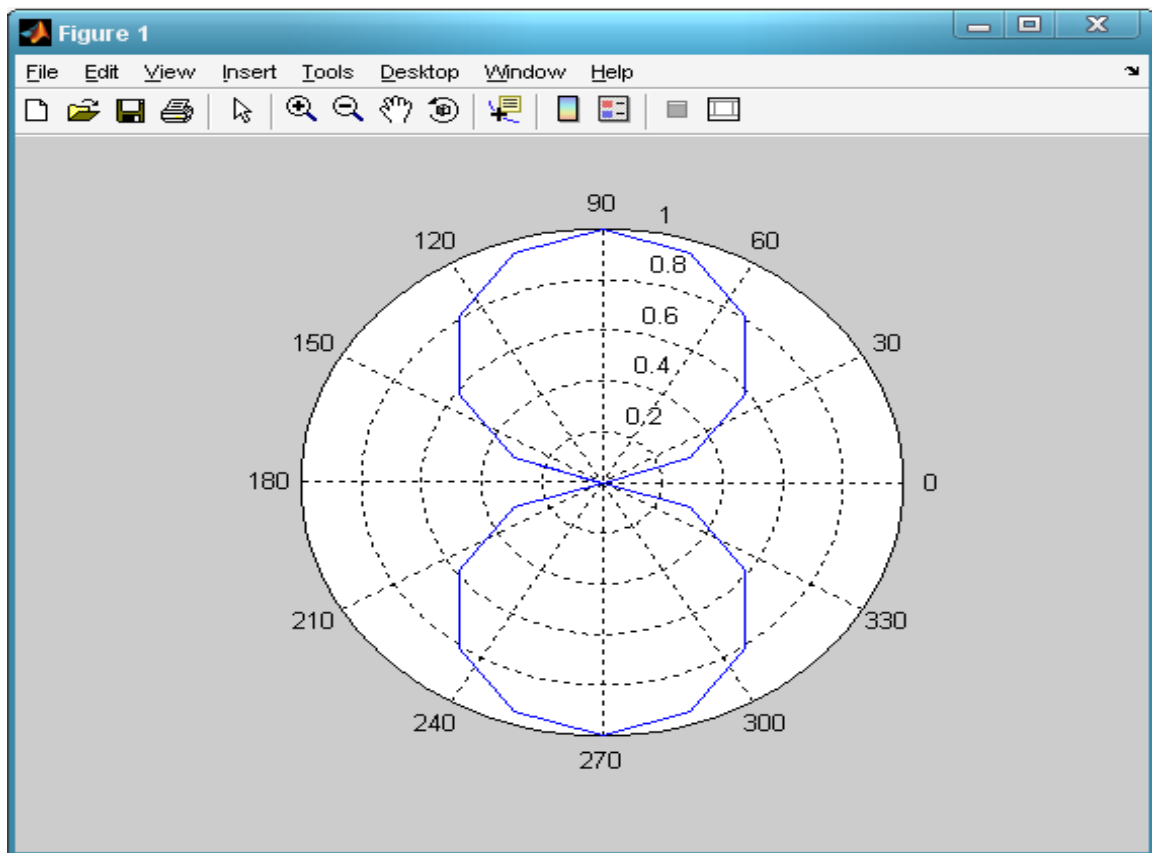
- **polyar(theta,rho)**- polyar koordinatalar tizimida radius-vektor oxirining o'z holatidagi rho uzunlik bilan theta burchakni ko'rsatuvchi grafikni quradi;
- **polyar(theta,rho,s)** - avvalgi buyruqdagi kabi , lekin s qatorli konstanta yordamida qurish uslubini (plot komandasiga o'xshash) beradi

Quyidagi misolni ko'ramiz:

```
>> angle=0:.1*pi:3*pi;  
>> r=exp(angle/10);  
>> polyar(angle,r),...  
>> title('poylar koordinatasida grafik');  
>> grid on
```



12.8 - rasm. Polyar koordinatalar tizimida grafik.



12.9 - rasm. Murakkab funksiya grafigi.

Yuqoridagi grafik (12.9-rasm) quyidagi komandalar yordamida chizildi:

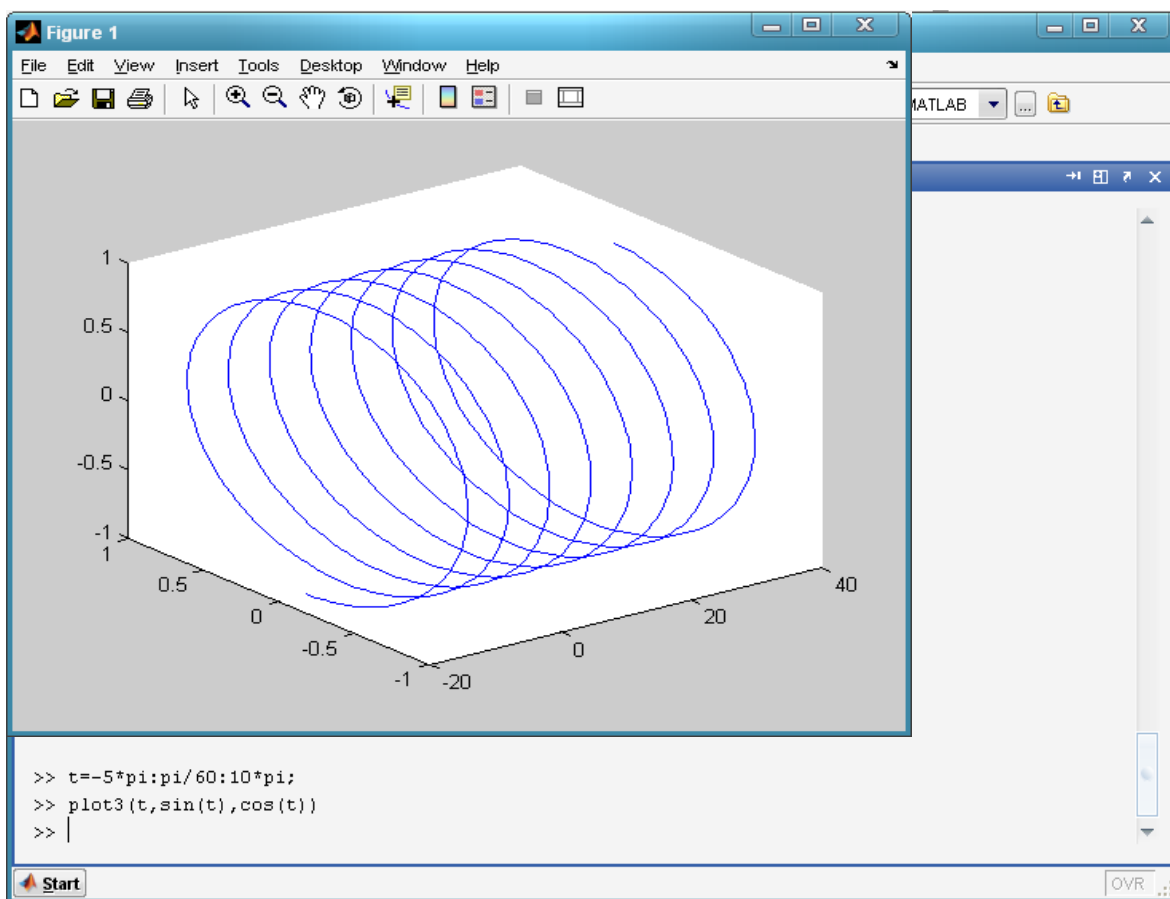
```
>> angle=0:.1*pi:3*pi;
>> r2=abs(sin(angle));
>> polyar(angle,r2)
```

12.4. Uch o'lchovli grafika

Uch o'lchovli fazoda chiziqning grafigini chizish uchun **plot3(x,y,z)** buyrug'idan foydalaniladi. Bunda x,y,z lar bir xil son da gi koordinatalarga ega bo'lgan Matlab vektorlari bo'lishi kerak. Masalan,

```
>> t=-5*pi:pi/60:10*pi;
>> plot3(t,sin(t),cos(t))
```

komandalar ketma-ketligi fazoda prujinasimon egri chiziqni grafigini chizadi.



12.10 - rasm. Uch o'lchovli fazoda chiziqning grafigi.

Bundan tashqari, har xil turdagi sirlarni hosil qilish uchun quyidagi komandalardan foydalanish mumkin:

- **mesh**- bu uch o'lchovli sirtni "to'r" sifatida chizadi;
- **surf**- uch o'lchovli sirt;

meshgrid funksiyasi yordamida x, y larning qiymatlaridan foydalanib, x, y matritsalar hosil qilinadi. Agar x, y larning qiymatlari bir xil to'plamda bo'lsa, **meshgrid** funksiyaning argumentida 1 ta argument qiymati ko'rsatilsa yetarli; x, y larning qiymatlari har xil to'plamda o'zgarsa, **meshgrid** funksiyaning argumentida ikkita to'plam ko'rsatiladi.

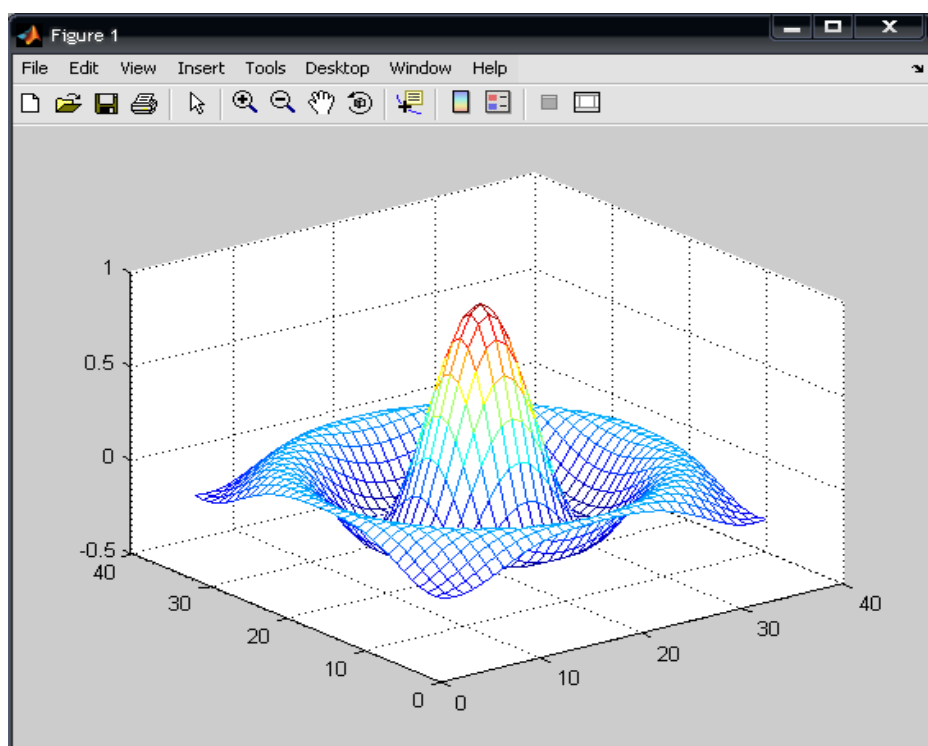
Masalan, 1) $Z = \sin R / R$, $R = \sqrt{x^2 + y^2}$, $x, y \in [-8, 8]$ bo'lsin, u holda

```
>> [x,y]=meshgrid(-8:.5:8); R=sqrt(x.^2+y.^2)+eps;
```

```
>> z=sin(R)./R;
```

```
>> mesh(z)
```

buyruqlar ketma-ketligi 12.11-rasmdagi sirtning grafigini fazoda chizib beradi:



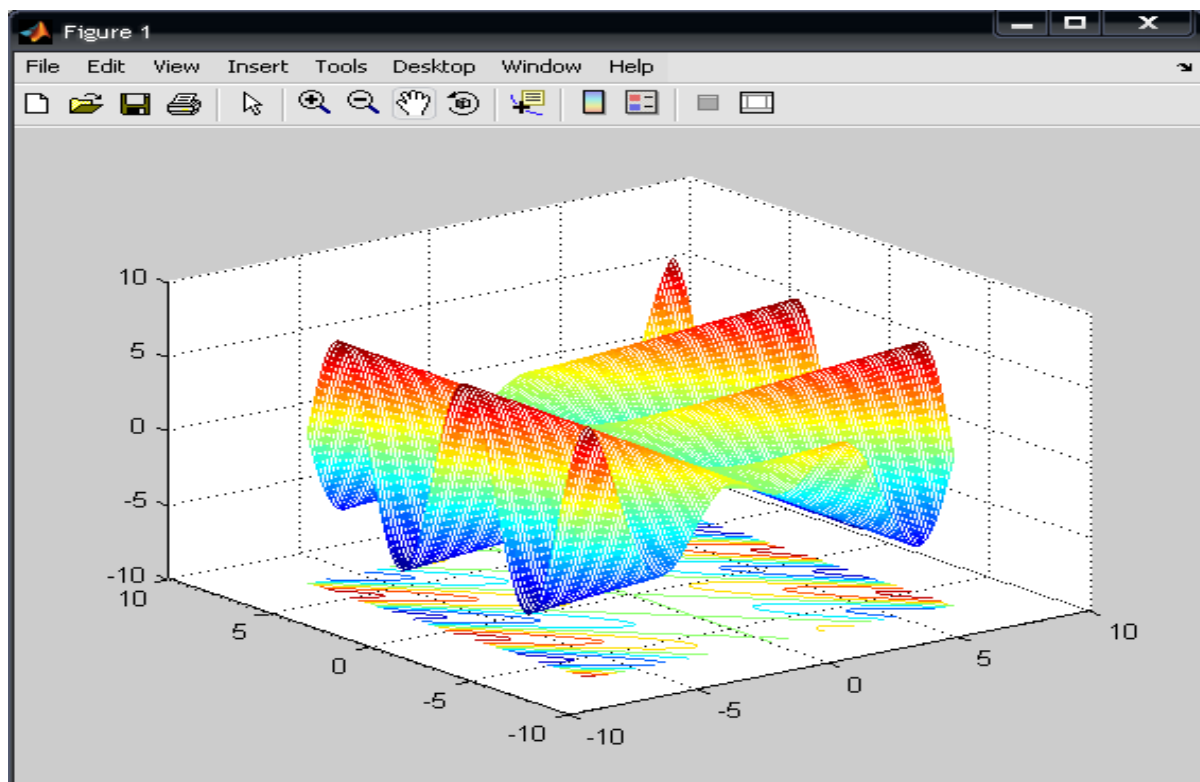
12.11- rasm. Sirtning to'rtli grafigi.

2) Sirtning soyali grafigi esa

```
>> [x,y]=meshgrid(-7:0.1:7); >> z=x.*sin(x+y);
```

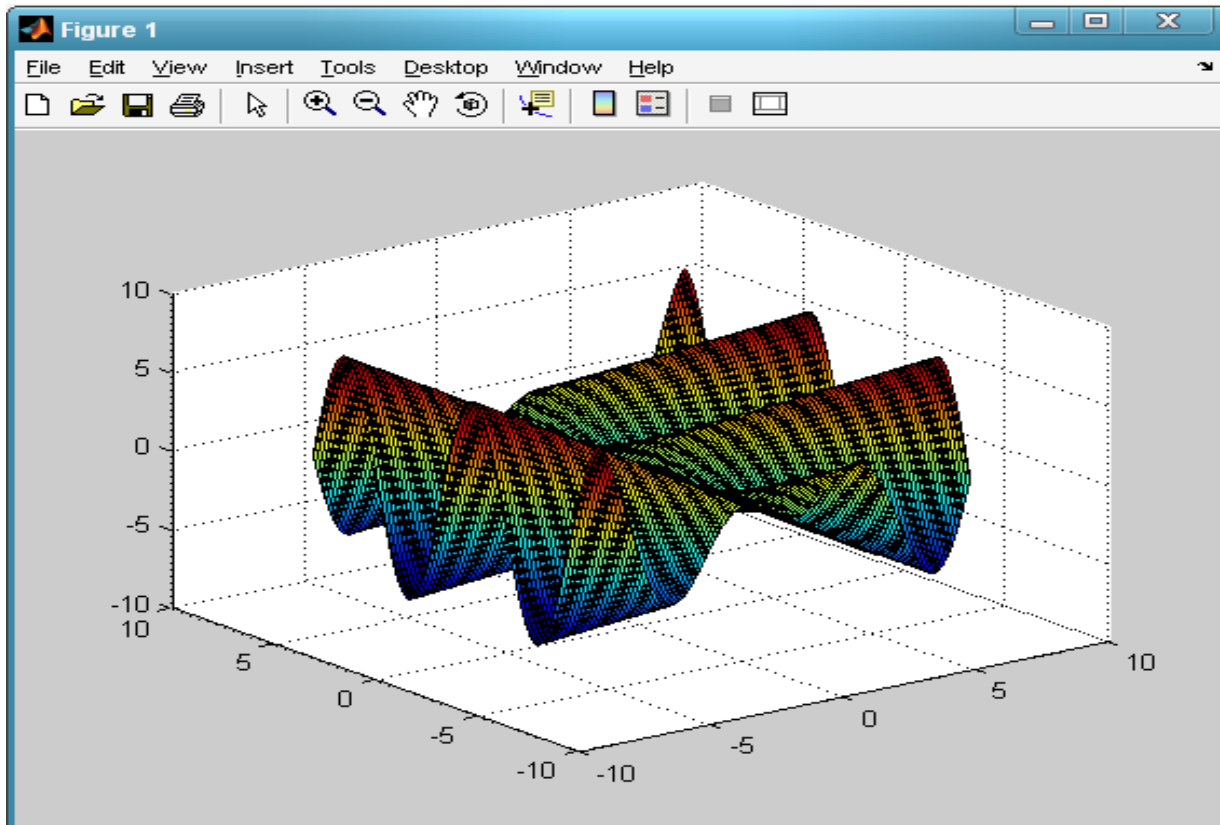
```
>> meshc(x,y,z)
```

 kabi komandalar yordamida chiziladi:



12.12-rasm. Sirtning soyali grafigi.

3) `>> surf(x,y,z)` komandasi esa quyidagi sirtni chizadi:

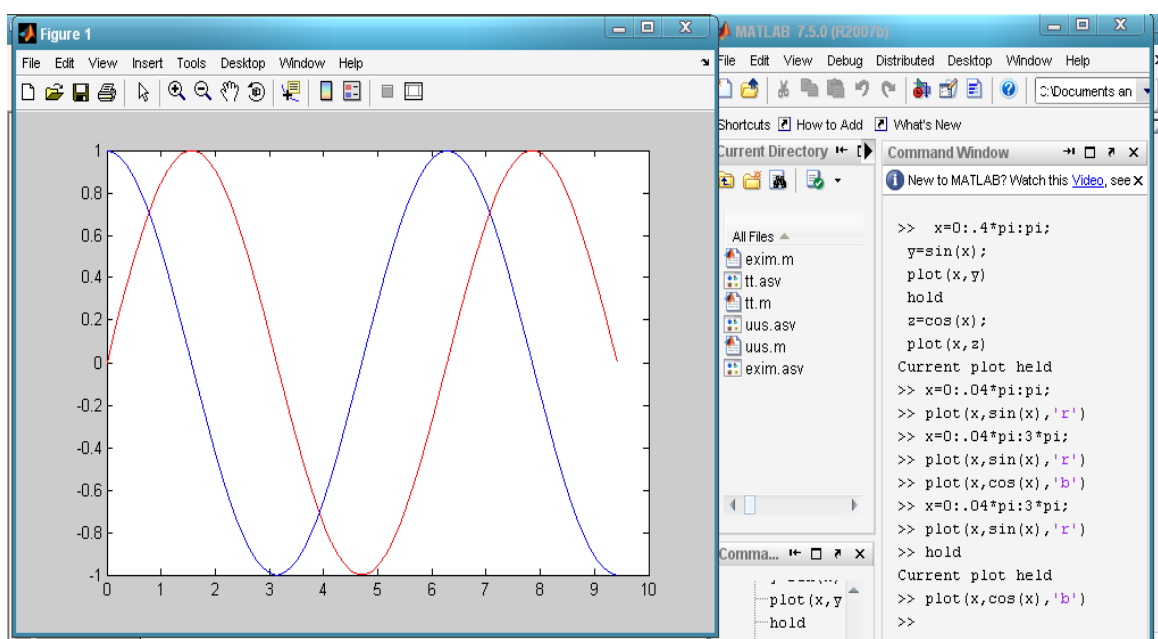


12.13 - rasm. Sirtning grafigi.

12.5. Bir nechta grafiklarni hosil qilish

Matlabda bir grafik oynasida bir necha grafiklar hosil qilish mumkin. Buning uchun grafik darchasini ochiq holda saqlash kerak. Bu esa **hold** buyruq'i yordamida amalga oshiriladi. Masalan, $y=\sin(x)$, $z=\cos(x)$, $x \in [0, \pi]$, funksiyalar grafigini bir oynada chizish uchun quyidagicha buyruqlar ishlatiladi (grafiklar 12.14 - rasmda):

```
>>x=0:pi/60:pi; y=sin(x); z=cos(x);  
>>hold  
>>plot(x,y,'b')  
>> plot(x,z,'r')
```



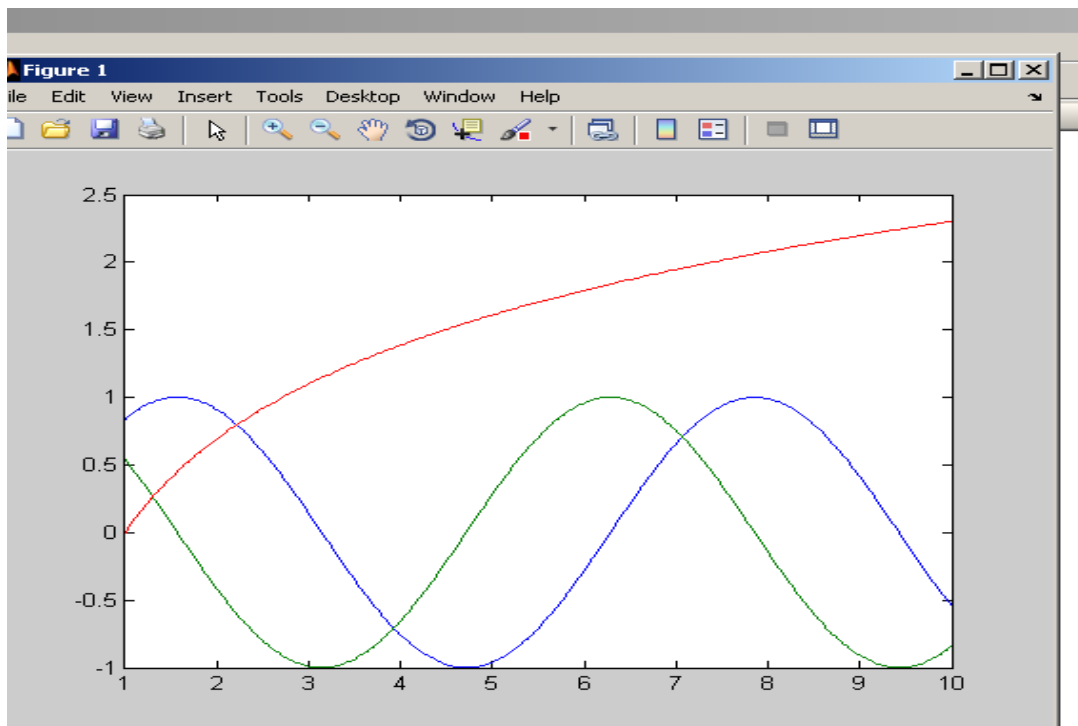
12.14 - rasm. Bir oynada ikkita grafik.

Shundan keyin, **hold off** komandasini hold ni ishlashini to'xtatuvchi sifatida ishlatish mumkin. Bir nechta grafikni bir oynada hosil qilish uchun quyidagi komandadan foydalansa ham bo'ladi : **plot(x,y1,x,y2,x,y3)**.

Misol. Komandalar oynasida yozilgan quyidagi ketma-ketlik

```
>>x=1:0.03;10;  
>>plot(x,sin(x),x,cos(x),x,log(x))
```

uchta funksiyaning grafigini bir oynada chizib beradi (12.15- rasm.):



12.15 - rasm. Bir oynada uchta grafik.

12.6. Silindr va sferani qurish

Matlabda uch o'lchovli fazoda silindrni grafisini hosil qilish uchun quyidagi maxsus komandalardan foydalaniladi:

1) $[x,y,z]=\text{cylinder}(R,N)$ - x,y,z massivlarni hosil qilib beradi. Bu massivlar yordamida R radiusli N ta tugun nuqtalardan iborat bo'lgan silindrni hosil qilish uchun zamin yaratadi. Shundan so'ng silindrni qurish uchun $\text{surf}(x,y,z)$ buyrug'i ishlatiladi.

2) $[x,y,z]=\text{cylinder}(R)$ yoki $[x,y,z]=\text{cylinder}$ - huddi yuqoridagi kabi bo'lib, bunda $R=[11]$, $N=[20]$.

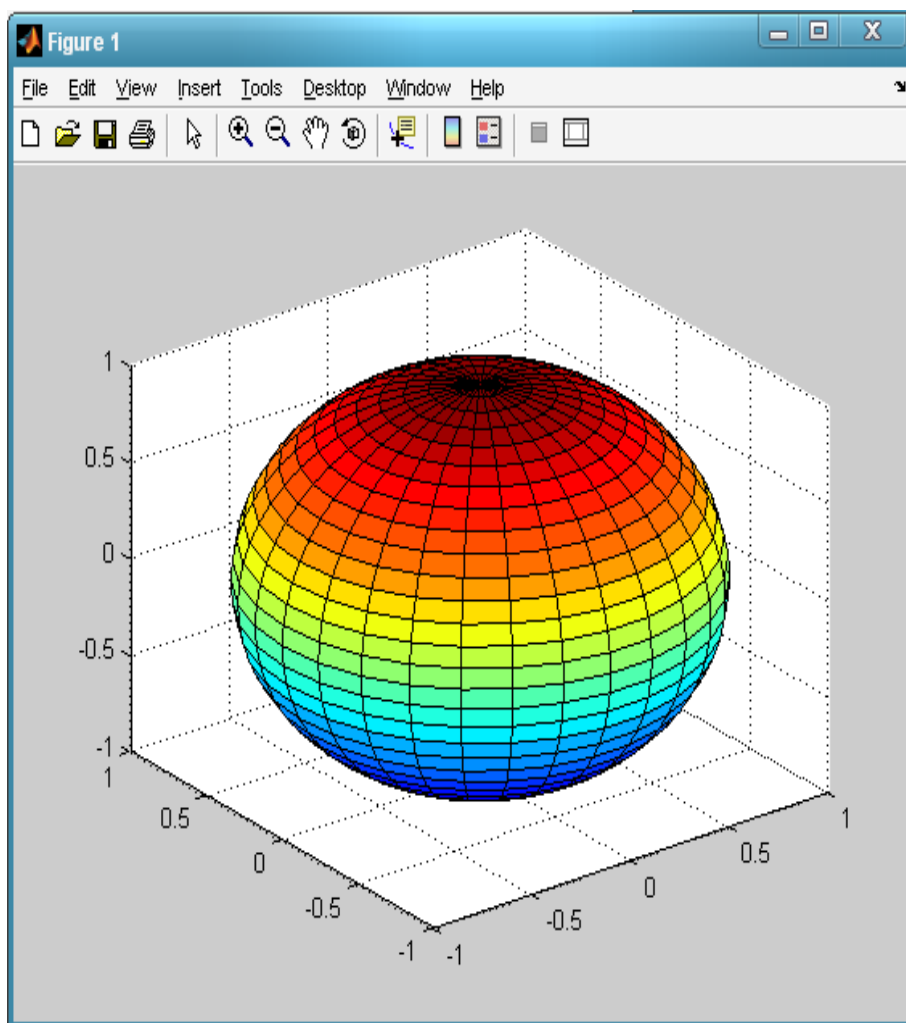
Sfera nuqtalarini aniqlash uchun **sphere** funksiyasi ishlatiladi. Uning formatlari quyidagicha:

1) $[x,y,z]=\text{sphere}(N)$ - $[-1,1] \times [-1,1] \times [-1,1]$ fazoda x,y,z -massivlar hosil qiladi. Ular $(N+1) \times (N+1)$ o'lchovli bo'ladi. Sfera qurish uchun $\text{surf}(x,y,z)$ yoki $\text{surfl}(x,y,z)$ komandasi ishlatiladi.

2) $[x,y,z]=\text{sphere}$ - huddi avvalgidek, faqat $N=20$.

Misol.>> $[x,y,z]=\text{sphere}(30)$;

>> $\text{surf}(x,y,z,x)$.

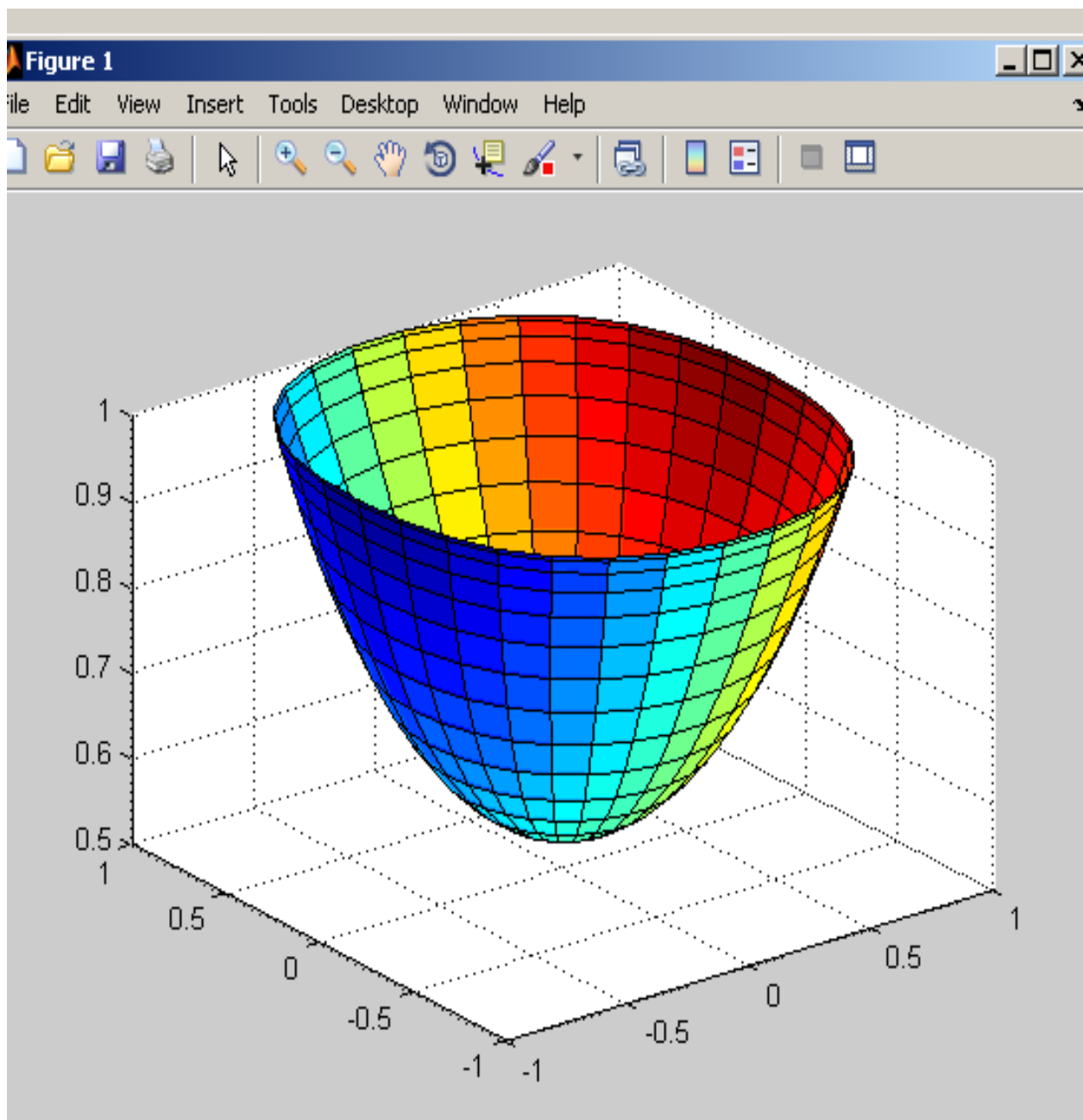


12.16 - rasm. Sfera grafigi.

Bunda yorug'lik effekti vektor rangi surf ning oxirgi argumenti x bilan berilyapti (12.16-rasm.), buni y yoki z bilan ham berish mumkin.

Shuni ta'kidlash lozimki, surf komandasi argumentlarini ifoda qilib bersa ham bo'ladi. Masalan (12.17- rasm.):

```
>>[x,y,z]=sphere(30);
>>surf(sin(x),y,cos(z),x)
```

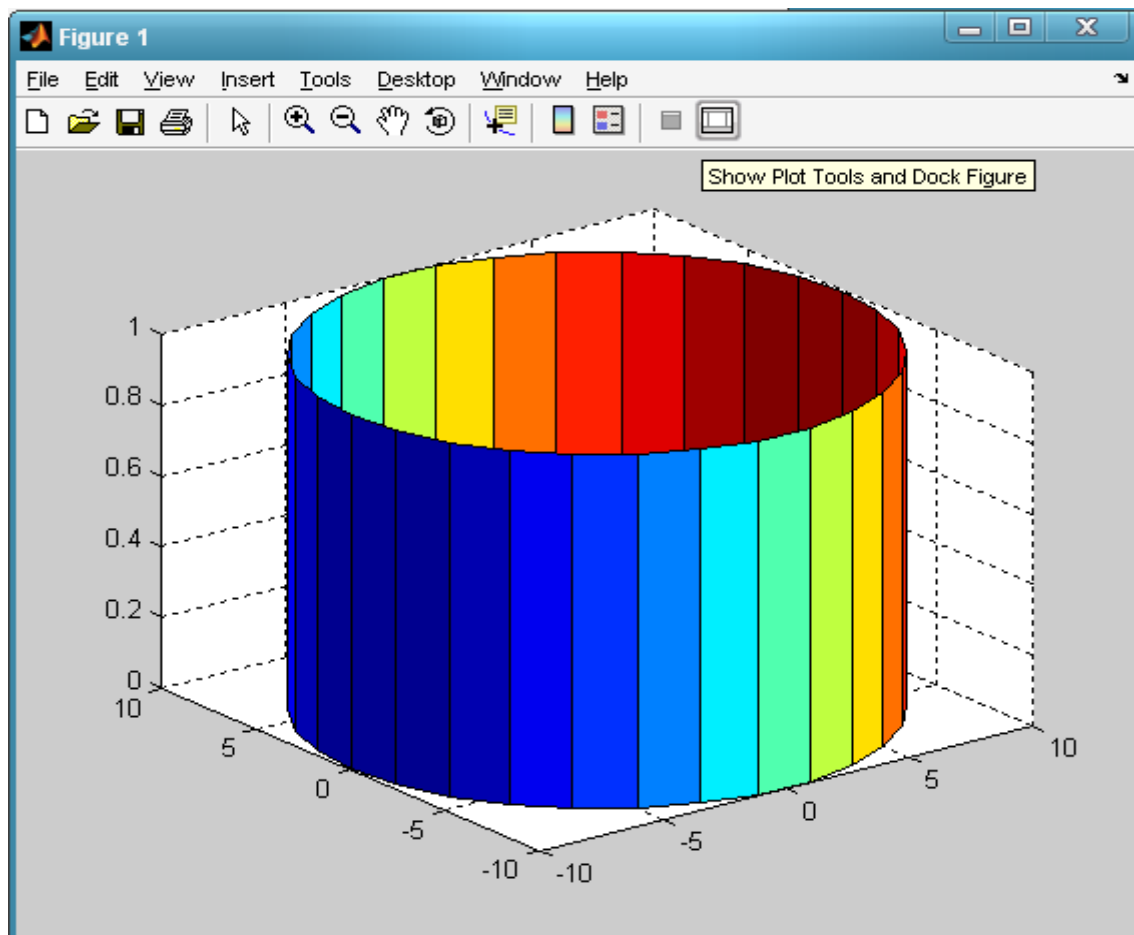


12.17 - rasm. Murakkab sirt grafigi.

Endi silindr grafiklarini chizishga doir misollar ko'rib chiqamiz.

```
1) >>[x,y,z]=cylinder(10,30);
>>surf(x,y,z,x).
```

Bu erda ham sferadagi kabi surf buyrug'i oxirgi argument x vektor orqali aniqlanuvchi rangga funksional buyoq berish imkoniyatini beradi.



12.18-rasm. Silindrning grafigi.

Ko'rinib turibdiki, silindr balandligi 1 ga teng.

2) Radiusi 4 ga teng bo'lgan silindr ichiga sferani joylashtiring.

```
>> [x,y,z]=cylinder(4,30);
```

```
>> surf(x,y,z,x);
```

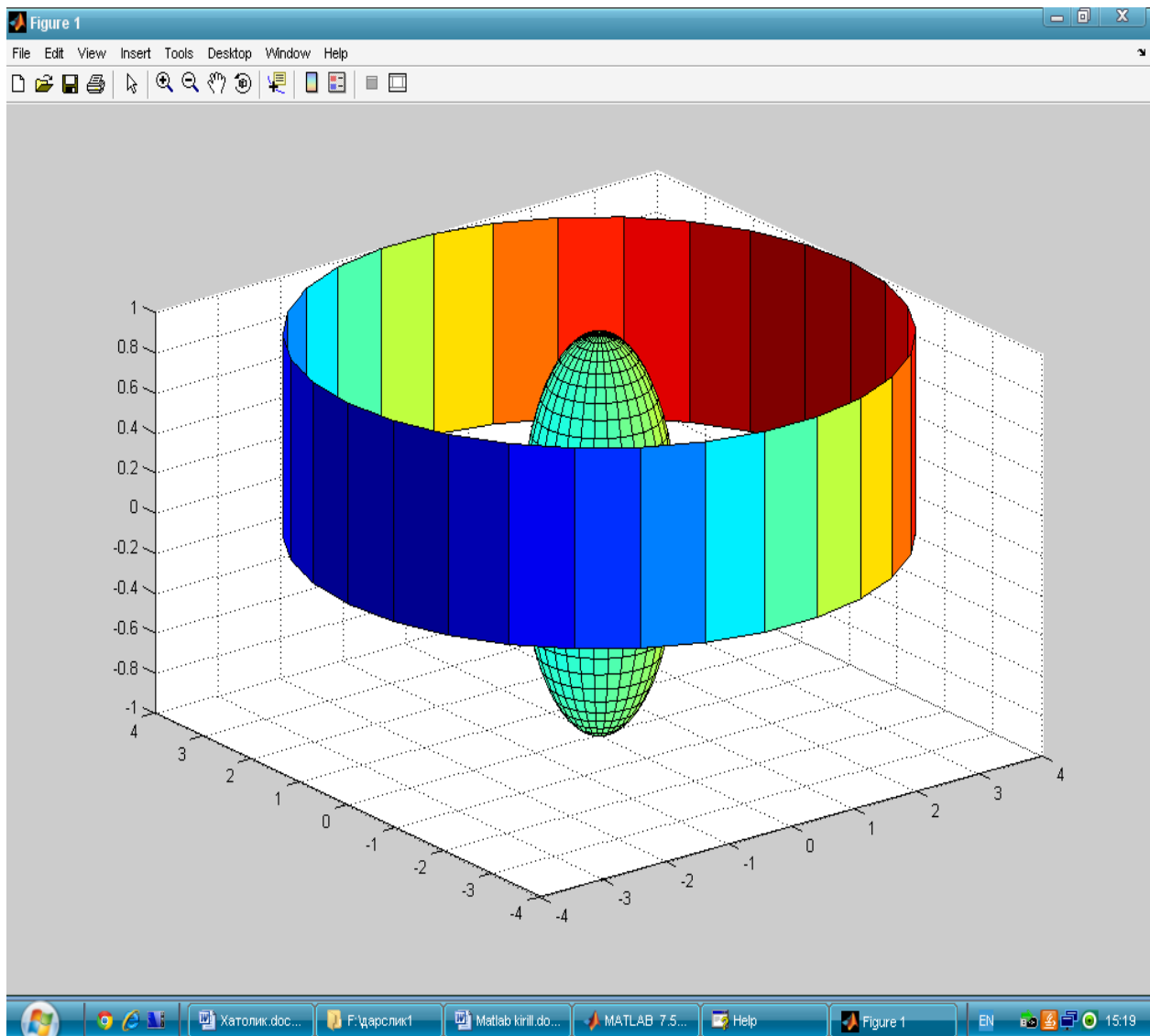
```
>> hold
```

Current plot held

```
>> [x,y,z]=sphere(30);
```

```
>> surf(x,y,z,x);
```

komandalar ketma-ketligi quyidagi grafiklarni chizadi:



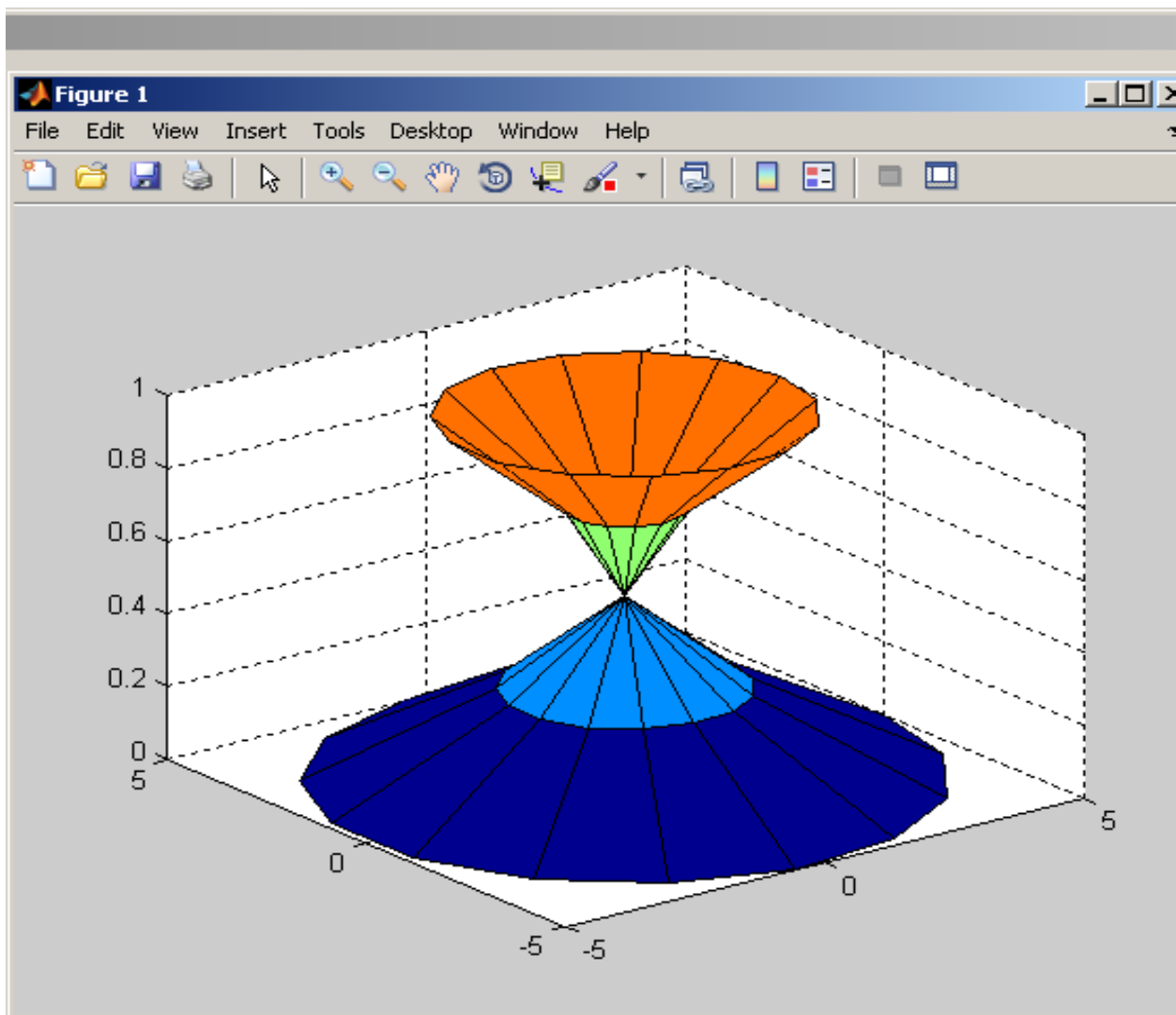
12.19 - rasm. Silindr va sfera grafiklari.

Silindr uchun yozilgan formatda R radius vektor yoki matritsa ham bo'lishi mumkin. Masalan,

```
>>[x,y,z]=cylinder([5 2 0 1 3],15);
```

```
>> surf(x,y,z)
```

buyrug'lari quyidagi shaklni hosil qiladi:



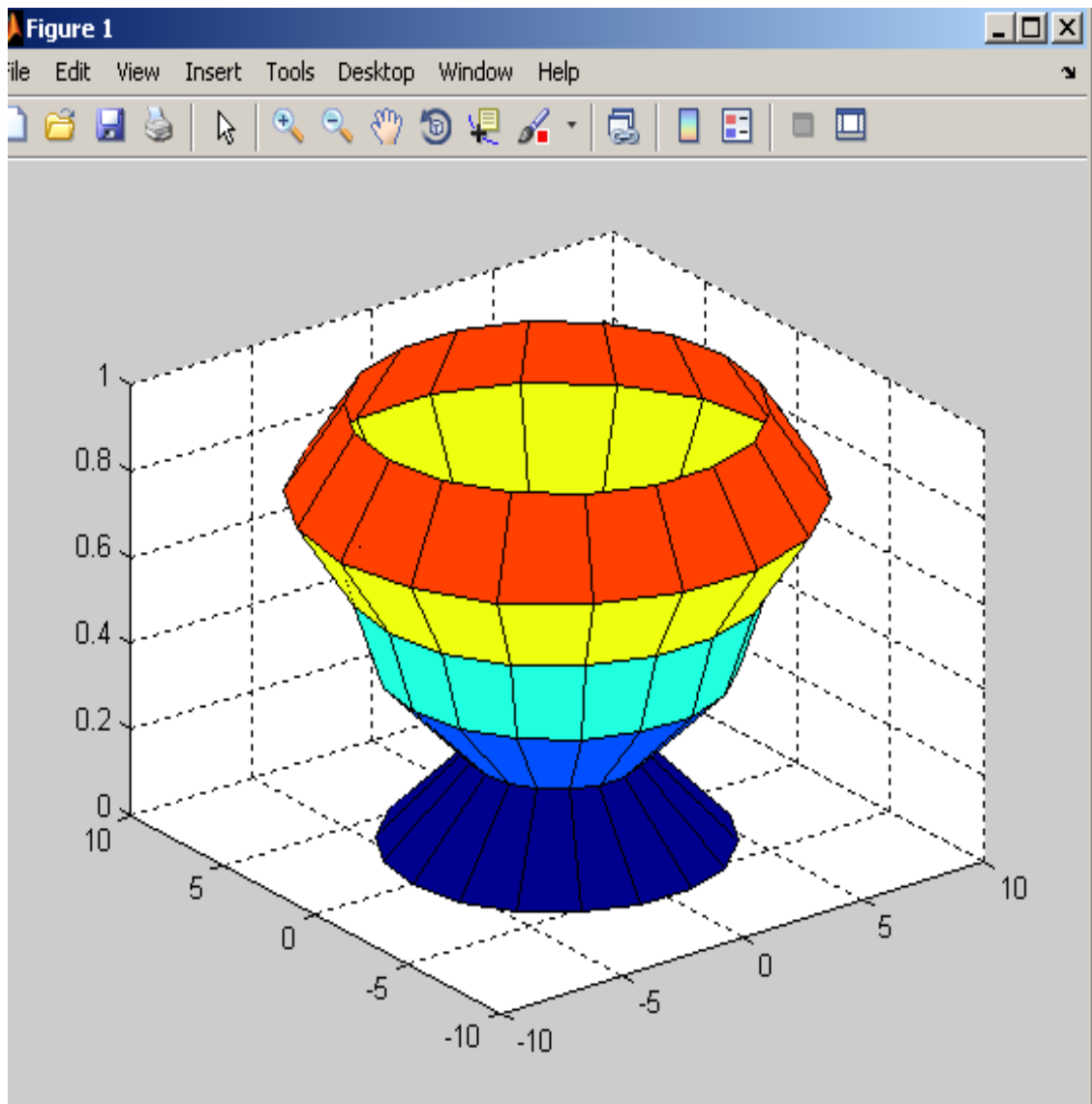
12.20 - rasm. Murakkab silindr grafigi.

E'tibor bering, z o'qining $[0 \ 1]$ kesmasi R vektorning elementlari soniga teng bo'linib, har bir qatlamda yani xOy tekisligiga parallel tekislikda o'ziga mos elementga teng radiusli aylanada ko'rsatilgan tugun nuqtalar birlashtiriladi.

Endi R massiv bo'lgan holni ko'raylik:

```
>>[x y z]=cylinder([6 7;3 9;6 7],18);
```

```
>>surf(x,y,z)
```



12.21 -rasm. Vektor formatli grafik.

Ta'kidlash joizki, tugunlar soni N qancha katta bo'lsa, shakl shunchalik silliqlashadi.

Nazorat savollari

1. Matlabda grafik chizishning qanday imkoniyatlari mavjud?
2. $\text{Loglog}(x,y)$ qanday grafikni yasaydi?
3. Gistogrammalar qanday buyruq yordamida quriladi?
4. Polyar koordinatalarda grafikni qurishga doir misol keltiring.
5. Meshgrid qanday funksiya?
6. Bir oynada bir nechta grafiklar qanday hosil qilinadi?
7. Silindrni uch o'lchovli fazoda qanday quriladi?
8. Sfera yasash uchun qaysi funksiya foydalaniladi?

Mustaqil ishlash uchun misollar

1. Matlab komandalari yordamida $y=\sin x+\cos x$ funksiya grafigini $[0; 4\pi]$ oraliqda chizing.

2. $X=[1 \ 2 \ 4, \ 7 \ 3 \ 5 \ 7]$ vektor gistogrammasini chizing.

3. $Y=[0.5 \ 2 \ 3.5 \ 4]$ vektor gistogrammasini $x=[1 \ 3 \ 4 \ 5]$ vektor elementlariga mos qilib chizing.

4. $f(x) = \sum_{i=1}^5 \left(\frac{1}{1+2ix}\right) - \sum_{i=1}^4 (1/(1+3ix))$ funksiya grafigini $x \in [0; 5]$, $x \in [1; 11]$ oraliqlarda alohida qilib chizing.

5. $Z=(1-x^2)(1-x^3)\sin(x+x^4)$ funksiya nolining taqribiy qiymatini grafik chizish yordamida aniqlang ($x \in [0; +\infty)$).

6. $\sin x$ ni logarifmini $\cos x$ ga nisbatan grafigini $x \in [\pi/10; 9\pi/10]$ oraliqda chizing.

7. $\sin x$ ni grafigini $\cos x$ ni logarifmiga nisbatan $x \in [-2\pi/5; 2\pi/5]$ oraliqda chizing.

8. $Y=\sin x$, $y=x*\cos x$, $x \in [0; 3\pi]$ funksiyalar grafiklarini polyar koordinatalar sistemasida chizing.

9. $z = \frac{\cos \sqrt{x^2 + y^2}}{1 + \sqrt{x^2 + y^2}}$ sirt grafigini $x, y \in [-7; 7]$ va $x, y \in [0; 14]$

oraliqlarda alohida qilib chizing (surf yordamida).

10. $Z=(x^2/2)-(y^2/2)$, $x, y \in [-5; 5]$ sirt grafigini mesh va surf komandalari yordamida chizing.

11. `magic(5)` matrisa gistogrammalarini tekislikda va fazoda chizing.

12. Quyidagi ifodalar bilan berilgan sirtlarning grafiklarini chizing, oraliqlarni mustaqil tanlang.

a) $Y = \ln(3 + 2\cos X)$

b) $Z = X^2 - XY + Y^2$

c) $Z = Y - Y^2 - X + 6Y$

d) $Z = X^3 + 8Y^3 - 6XY + 1.$

13. MAXSUS GRAFIKA. ANIMATSIYA BAJARISH VOSITALARI

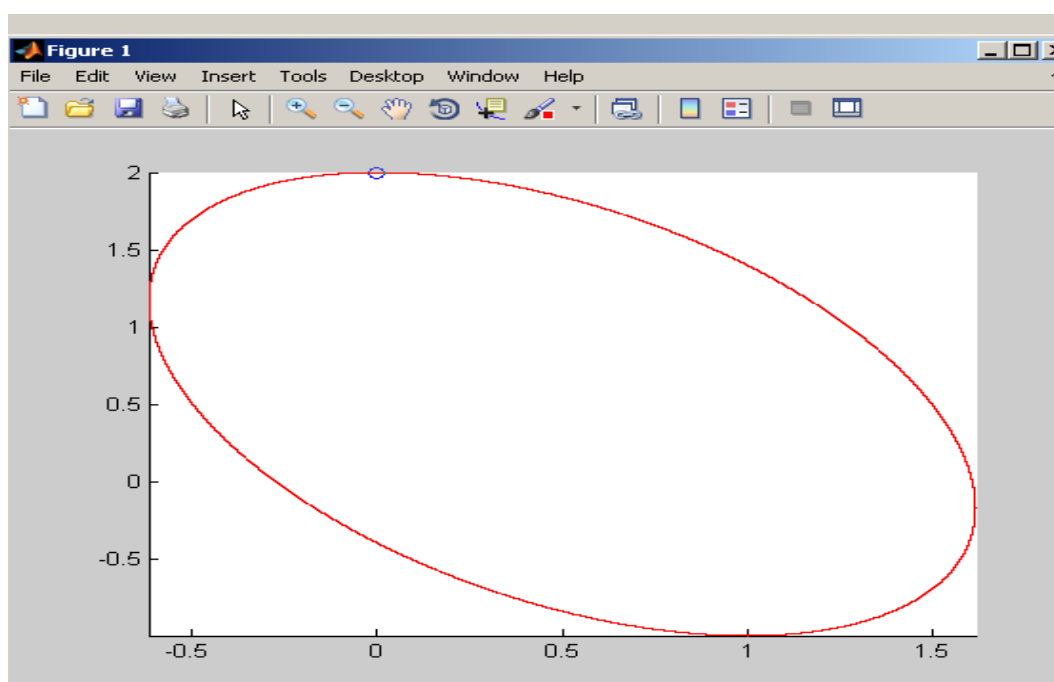
13.1. Animatsiyani bajarish vositalari

Nuqtaning tekislikda harakatlanish traektoriyasini aks ettirish uchun comet komandasidan foydalaniladi. Bunda nuqta izga ega bo'lgan kometaning yadrosini eslatadi. Ushbu komanda quyidagi ko'rinishlarda qo'llaniladi:

- **comet(y)** - “kometa”ning y vektor bilan berilgan traektoriya bo'yicha harakatlanishini aks ettiradi;
- **comet(x,y)** - “kometa”ning y va x vektorlar juftligi bilan berilgan traektoriya bo'yicha harakatlanishini aks ettiradi;
- **comet(x,y,z)** - avvalgi komandaga o'hshash, faqat kometa izining uzunligini ham ko'rsatish mumkin. Kometaning izi boshqa rangga bo'yalgan bo'ladi, u $p \cdot \text{length}(y)$ ko'rinishida beriladi ($\text{length}(y)$ - y vektorning o'lchami, $p < 1$, sukut bo'yicha $p=0,1$).

Quyida comet komandasidan foydalanishga doir misol keltirilgan:

```
>> t=0:.01:2*pi;  
>> y=sin(2*t)+(sin(t).^2);  
>> x =cos(2*t)+(cos(t).^2);  
>> comet(y,x,0.3);
```



13.1- rasm. Tekislikda nuqtaning harakati.

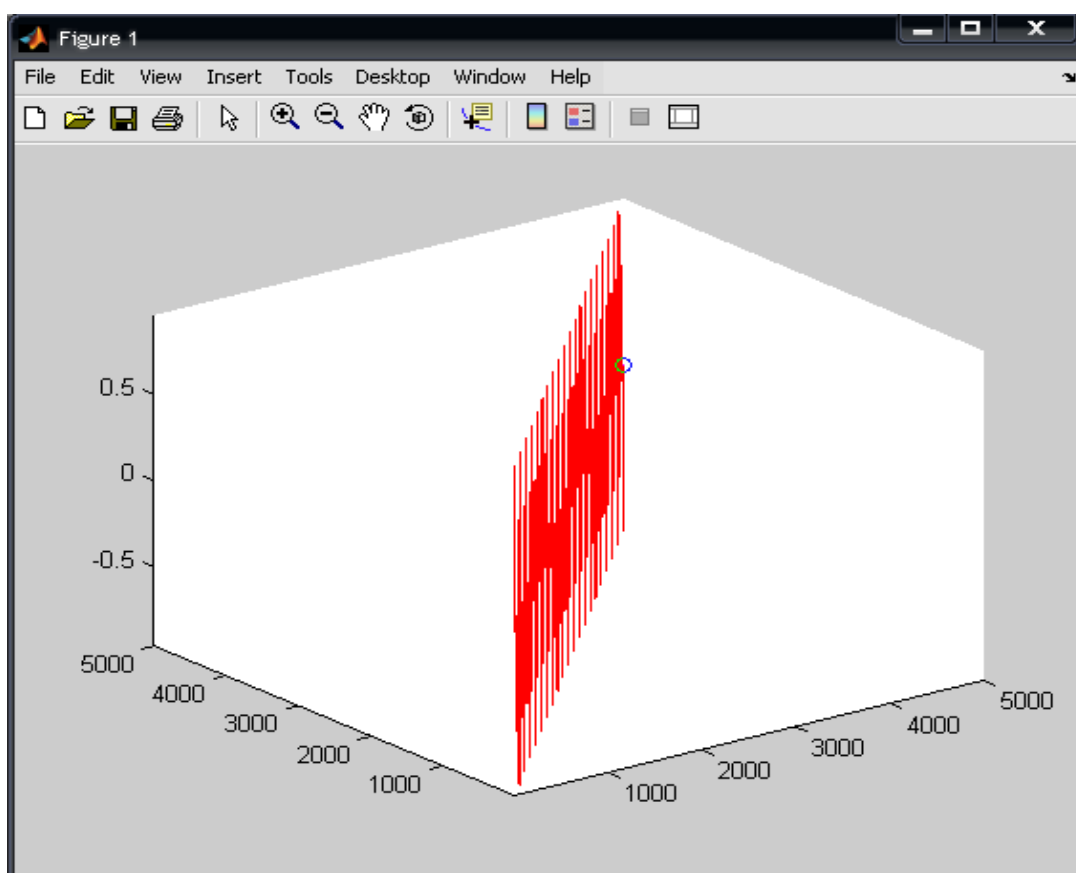
13.2. Nuqtaning fazoda harakatlanishi

Nuqtaning uch o'lchamli fazoda harakatlanishini kuzatish uchun quyidagi ko'rinishlarga ega bo'lgan **comet3** komandasidan foydalaniladi:

- **comet3(z)** - nuqtaning z vektor bilan berilgan uch o'lchamli egri chiziq bo'yicha harakatlanishini aks ettiradi;
- **comet3(x,y,z)** - nuqtaning fazoda $[x(i), y(i), z(i)]$ nuqtalar bilan aniqlanadigan egri chiziq bo'yicha harakatlanishini aks ettiradi;
- **comet3(x,y,z,p)** - avvalgi komandaga o'xshash, faqat nuqta izining uzunligini ham ko'rsatish mumkin. Nuqtaning izi $p * \text{length}(y)$ ko'rinishida beriladi ($\text{length}(y)$ -y vektorning o'lchami, $p < 1$, sukut bo'yicha $p=0,1$).

Quyida comet3 komandasidan foydalanishga misol keltirilgan:

```
>> t=-10*pi:pi/250:10*pi;  
>> z=(sin(5*t).^5).*cos(t);  
>> comet3(z);
```



13.2 - rasm. Fazoda nuqtaning harakati.

Nuqtaning ikki va uch o'lchamli fazodagi harakati eng sodda animatsiyalardan bo'lishiga qaramasdan dinamik masalalarni grafik vizuallashtirish imkoniyatlarini kengaytiradi.

13.3. Deskriptorli grafika

Deskriptorli grafika bilan tanishishdan avval, grafik ustida bajarilishi mumkin bo'lgan ba'zi yordamchi tushunchalarni o'rganamiz. Bulardan biri grafik chiziqlarni markerlash va formatlashtirishdir. Dekart tekisligida kursorni chiziq ustiga qo'yib sichqonchani chap tugmasini bosilsa, chiziq ustida uni xarakterlovchi qora kvadratchalar hosil bo'ladi va chiziq alohida ko'rinishga ega bo'ladi. Ma'lumki, dekart tekisligida grafik chiziqlari berilgan (x,y) juftlik nuqtalarni mos oraliqdagi o'rinlarni tutashtirish natijasida hosil qilinadi. Shu nuqtalar har xil belgilar (markerlar) bilan belgilanishi mumkin. Masalan, bu belgilar "o", "x", "*" va boshqalar bo'lishi mumkin. Grafik chiziqlar ustida markerlarni hosil qilganda ularni o'lchamlarini, rangini berish mumkin bo'ladi. Grafik chiziqlarda markerlarni ishlatish ularni alohida ajratib, ko'rinarliroq bo'lishini ta'minlaydi.

Undan tashqari quyidagi grafik oyna interfeysidan foydalanish mumkin:

- Copy Figure – grafikni buferga nusxalash;
- Copy Options- grafik parametrlarni nusxalash;
- Figure Properties- grafik xossalari oynasini chiqarish;
- Axes Properties- grafik o'qlari xossalari oynasini chiqarish;
- Surrent Object Properties – joriy ob'yekt xossalari oynasini chiqarish.

Deskriptor grafikasi degani – ochib beruvchi yoki handle-grafikani anglatadi. Bu grafika barcha grafik komandalarning va foydalanuvchi interfeysini ob'yektga yo'naltirilgan dasturlash bilan ta'minlab beradi.

Deskriptor grafikasining asosiy tushunchasi bo'lib grafik ob'yekt hisoblanadi. Uning foydalanish uchun zarur bo'lgan Tools mexanizmlar menyusi quyidagilardir:

- Edit Plot - grafikni tahrirlash;
- Zoom In - grafik masshtabni kattalashtirish;
- Zoom Out - grafik masshtabini kichiklashtirish;
- Rotate 3D -fazoda (uch o'lchovli) grafikni burish(aylantirish);
- Basic Fitting - approksimatsiya qilish;
- Data -grafik nuqtalari uchun statistik ma'lumotlarni olish;
- Rectangle - to'q'ri to'rtburchaklarni yaratuvchi ob'yekt;
- Surface - sirtni yaratuvchi ob'yekt;
- Text - tekstli yozuvlarni yaratuvchi ob'yekt;
- Light - yorug'lik effektini yaratuvchi ob'yekt.

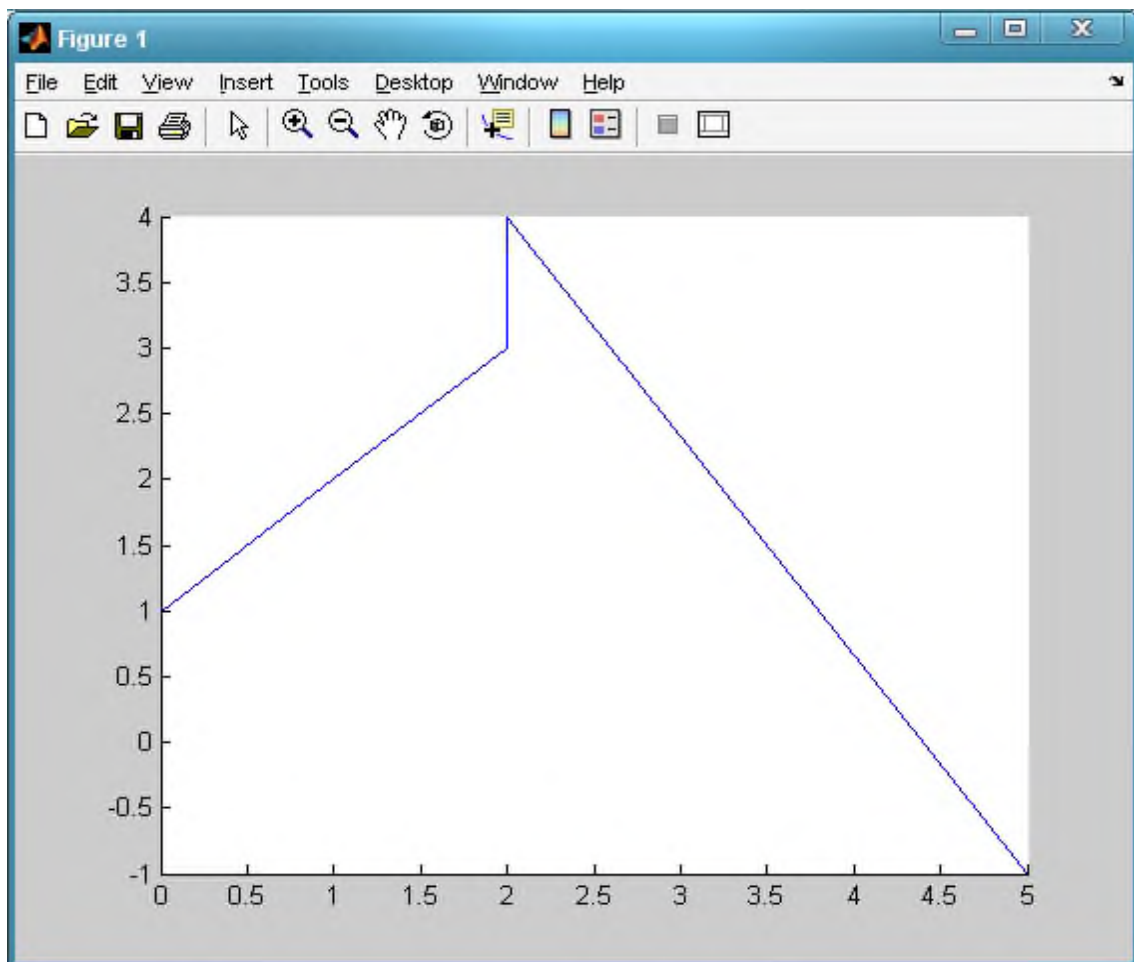
Ob'yektlar o'zaro boq'langandir va qandaydir grafik effektini hosil qilish uchun bir- biriga murojat qilishi mumkin.

Koordinata o'qlarini yaratish va boshqarish uchun quyidagi komandalar ishlatiladi:

- axes - koordinata o'qlarini yaratuvchi komanda;
- box - rasmni atrofida to'rtburchak qurish komandasi;
- cla - axes qurishlarni olib tashlash;
- hold -koordinata o'qlarini saqlab turish;
- ishold - hold statusini tekshirish(agar hold ishlayotgan bo'lsa, 1 ga teng , aks holda 0 ga teng).

Deskriptor grafikasi ob'yektini qo'llashga misol ko'raylik: (0,1),(1,2),(2,3) (2,4) va (5,-1) nuqtalardan o'tuvchi chiziq grafigi qurish talab qilinsin. Buning uchun line ob'yektidan foydalanamiz. Bu ob'yekt huddi shu nomdagi quyidagi grafik komandasi bilan quriladi:

```
>> line([0 1 2 2 5],[1 2 3 4 -1],'color','blue')
```



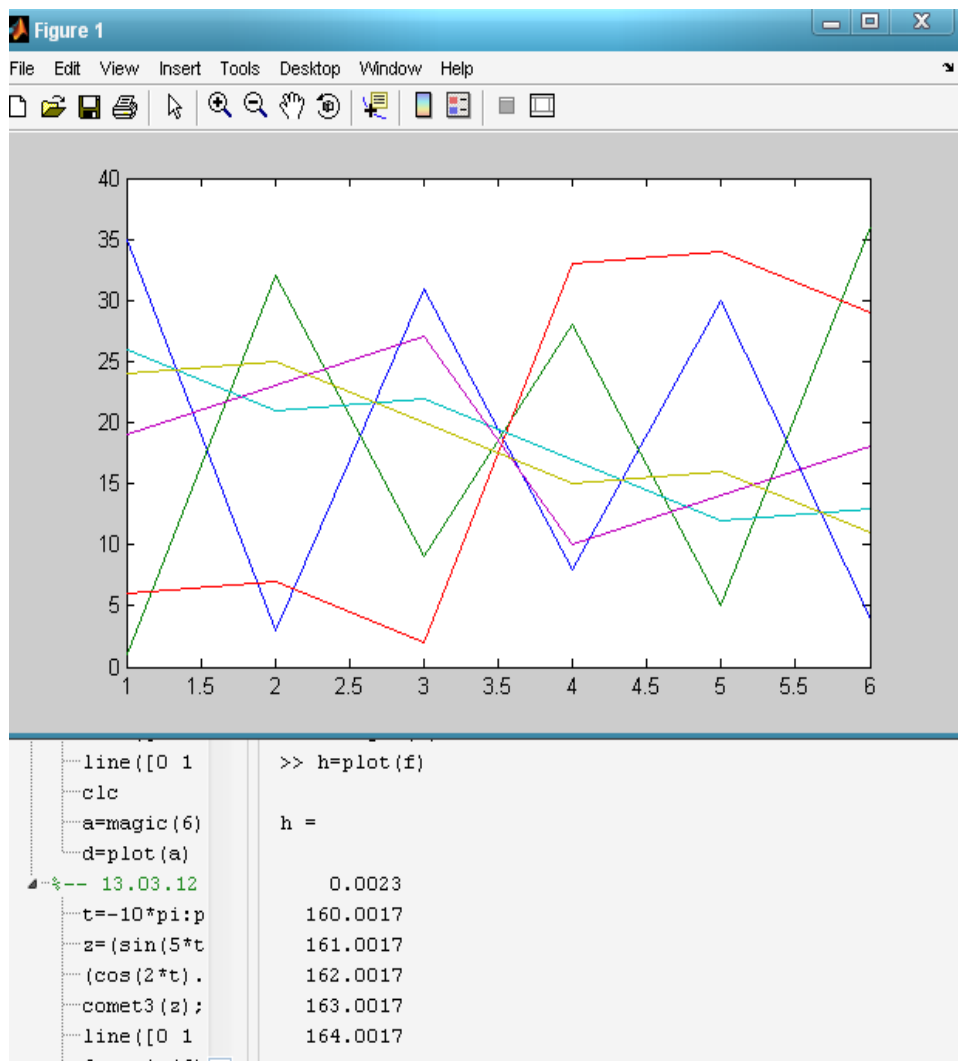
13.3 - rasm. Line yordamida chizilgan grafik.

Line komandasining xususiyatlari shundan iboratki, unda grafik qurishning barcha shartlari ochiq holda ko'rsatilgan bo'ladi. Bular yuqoridagi misolda konkret nuqtalar koordinatalari, rang parametrlari color va rangning o'zi 'blue'.

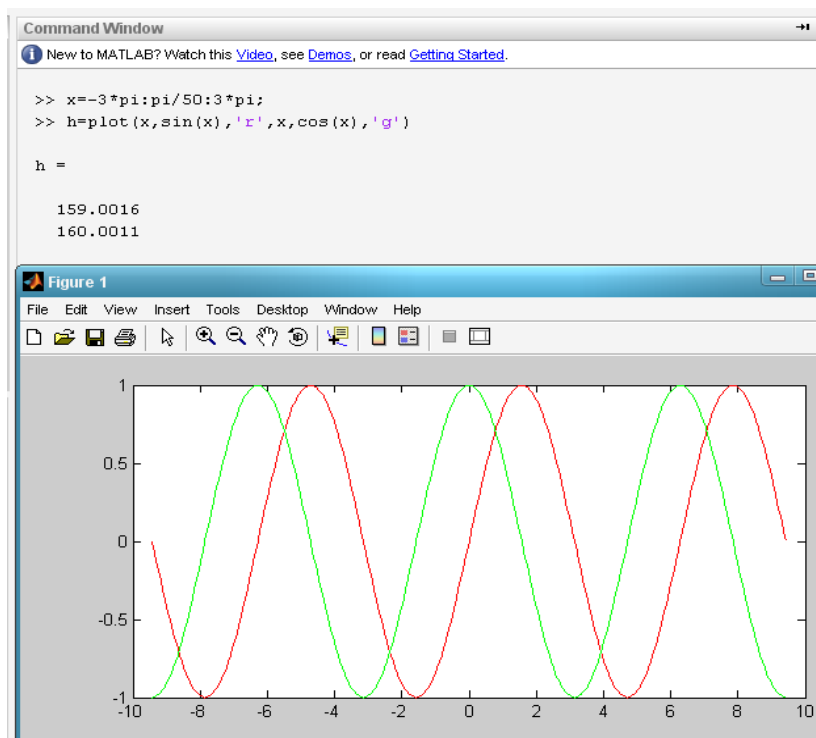
13.4. Ob'yektlar deskriptorlari

Deskriptor grafikasi ob'yektlari tushunchasi bilan ob'yektlarning alohida xususiyatini bildiruvchi deskriptor bo'q'langandir. Deskriptorni ob'yektlarni aniqlovchi qandaydir son deb tushunish mumkin. Masalan, root ob'yektlarining deskriptori har doim 0 ga teng, figure ob'yektlarini deskriptori grafik oynaning nomerini bildiruvchi butun son, boshqa ob'yektlarniki esa suzuvchi vergulli sonlardir. Bitta shunday ob'yektning deskriptori bitta son bo'ladi, bir necha ob'yektlarning deskriptori bir nechta sonlar(vektor) bo'ladi. Masalan,

1)



13.4 - rasm. Deskriptorli grafik.



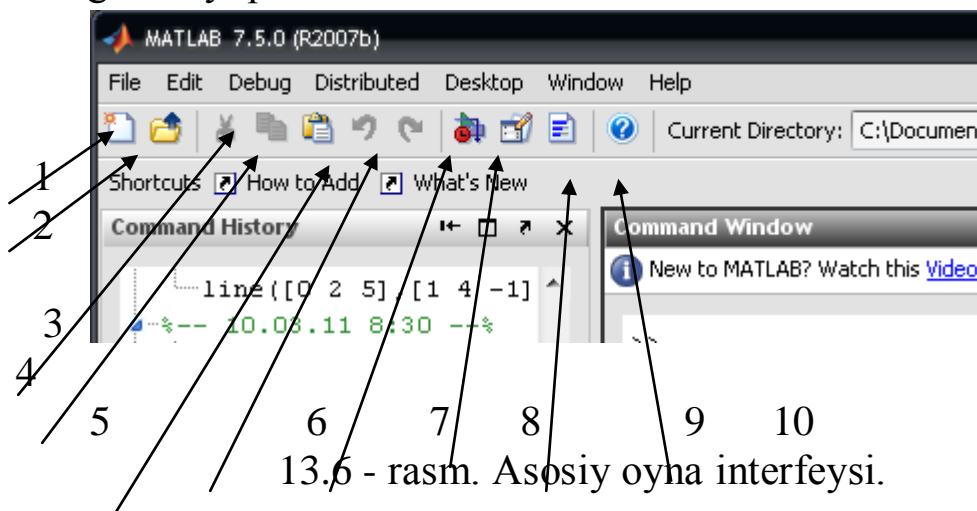
2)

13.5 - rasm. h - Figure ob'yektining deskriptori.

Yuqoridagi misollardan ko'rinib turibdiki, grafik chizuvchi plot komandasi yordamida yaratilgan figure ob'yektining deskriptori h vektor bo'lib, u birinchi misolda oltita koordinataga, ikkinchisida esa ikkita koordinataga ega.

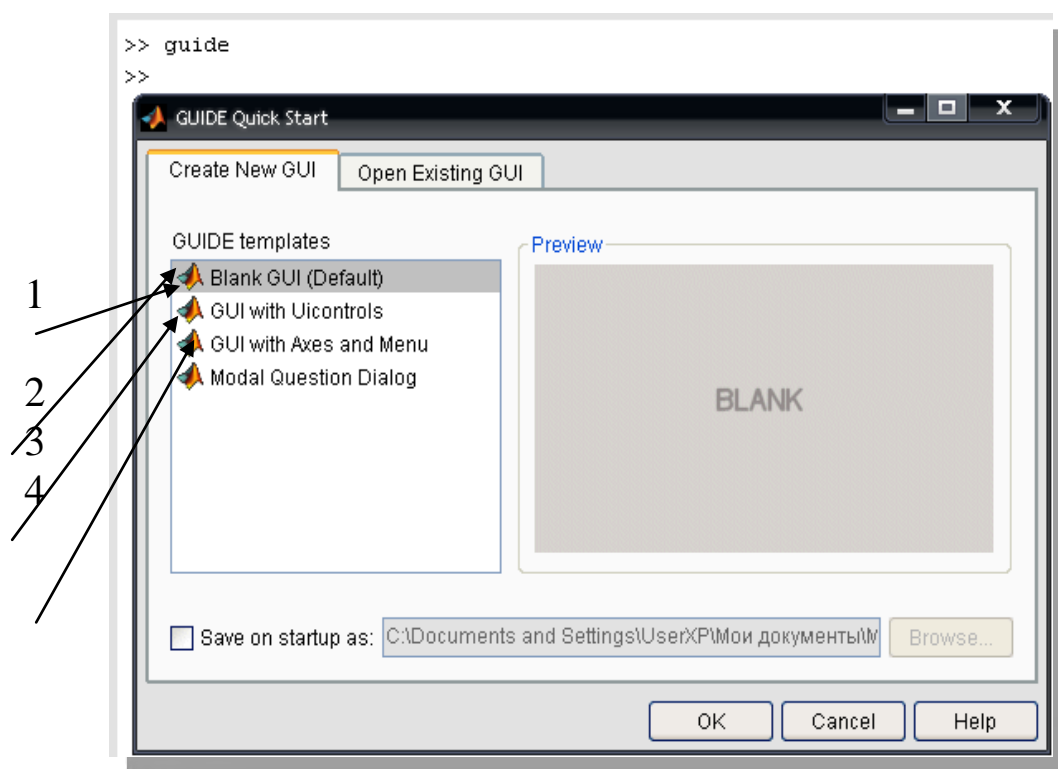
13.5. Foydalanuvchi interfeysini yaratish

Matlab tizimi bilan ishlash imkoniyatini foydalanuvchi interfeysi orqali amalga oshiriladi. Instrumentlar paneli oynasi Matlab dasturi oynasining asosiy qismlardan biri:



13.6 - rasm. Asosiy oyna interfeysi.

1. New M_file (Новый m_файл) –Yangi m-faylni ochish;
2. Open file (Открыть файл) – m-faylni ochish;
3. Cut (Вырезать) – belgilangan qismni kesib oladi va buferga joylashtiradi ;
4. Copy (Копировать) –belgilangan buffer fragmentidan nusxa oladi;
5. Paste (Вставить) –chiqarish kerak bo'lgan qatorga buffer fragmentini joylashtiradi;
6. Undo (Отменить) –avvalgi operatsiyani bekor qilish;
7. Redo (Повторить) –bekor qilingan so'nggi operatsiyani qayta tiklash;
8. Simulink –Simulink kutubxonasidan brauzer oynasini ochish;
9. GUIDE – grafik interfeysli kengaytirilgan oynani ochish;
10. Help (Помощь) –ma'lumotnoma oynasini ochish.



13.7 - rasm. Maxsus oyna.

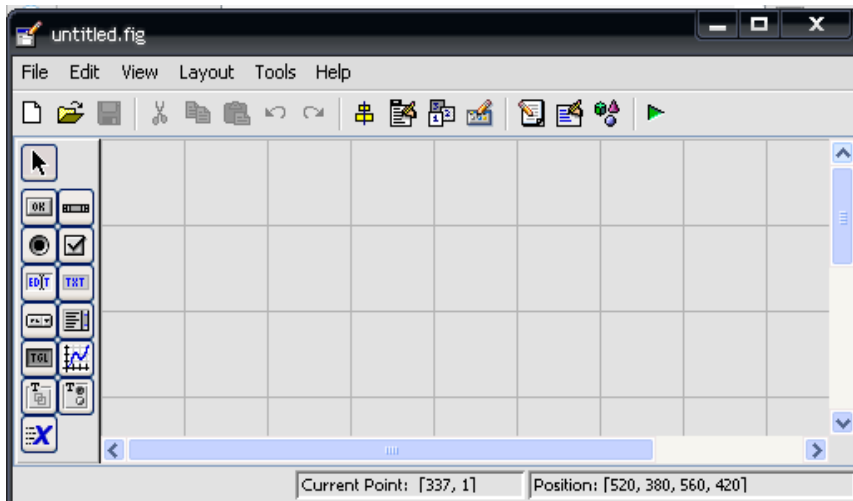
1. Standart uskunalar oynasi Blank GUI (default). Bunda barcha uskunalar va ob'yektlar passiv holatda bo'ladi. Bo'sh formaning o'zi mavjud bo'lib, kerakli uskunalarni foydalanuvchi o'zi o'rnatadi.

Matlabda kiritilayotgan buyruqlarni ko'rsatib borish vazifasi alohida grafik interfeysga yuklatiladi. Bu interfeys foydalanuvchi interfeysi deyiladi - GUI(Graphe User Interface). Ushbu dasturda boshqa yuqori

darajadagi ob'yektga yo'naltirilgan dasturlashda bo'lgani kabi bir nechta ma'lumotlarni kiritish vositalaridan foydalanish mumkin. Ularni ifodalash uchun an'anaviy grafik interfeysdan foydalanish yetarli emas. Buning uchun maxsus vizual grafik interfeys tashkil etilgan bo'lib, uning nomi GUIDE(Graphe User Interface Designer) deyiladi. Ushbu interfeys alohida kutubxona ko'rinishida tashkil etilgan bo'lib, uning tarkibiga barcha vizual ma'lumot kiritish ob'yektlari joylashtirilgan. Ularga misol sifatida tugma, checkbox, radio, matn kiritish ob'yekti, grafik chizish ob'yekti va boshqalarni olish mumkin. Ushbu kutubxonani ishga tushirish uchun foydalanuvchi ishchi stoliga quyidagicha buyruq beriladi:

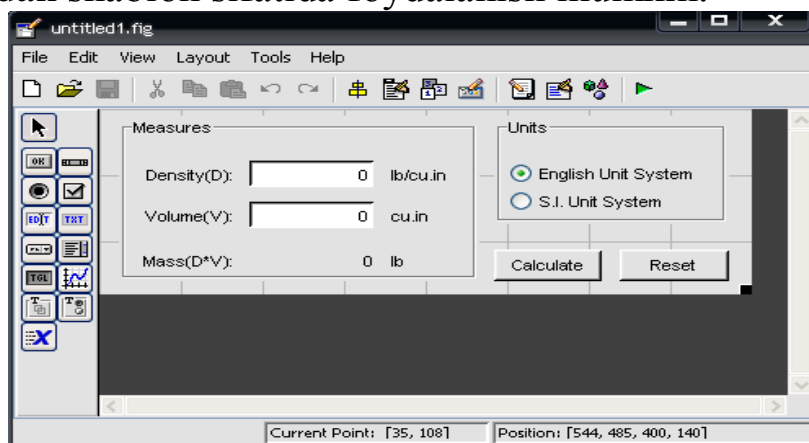
>>guide

Ushbu buyruq kiritilganda maxsus oyna ochilib, bu oyna yordamida foydalanuvchi o'ziga kerakli bo'lgan dizayndagi oynani tanlashi mumkin. Bu oynalarni 4 xil varianti bor:



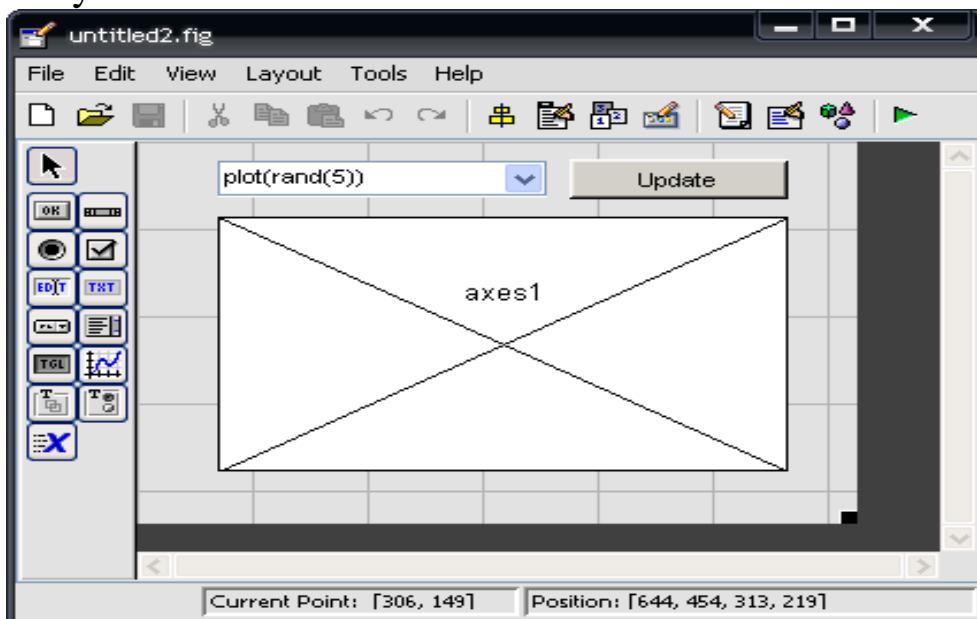
13.8 - rasm. Uskunalar va ob'yektlarning passiv holati.

2. GUI with Uicontrols. Ushbu bo'limda bir qancha ob'yektlar aktiv hisoblanib, undan shablon sifatida foydalanish mumkin.



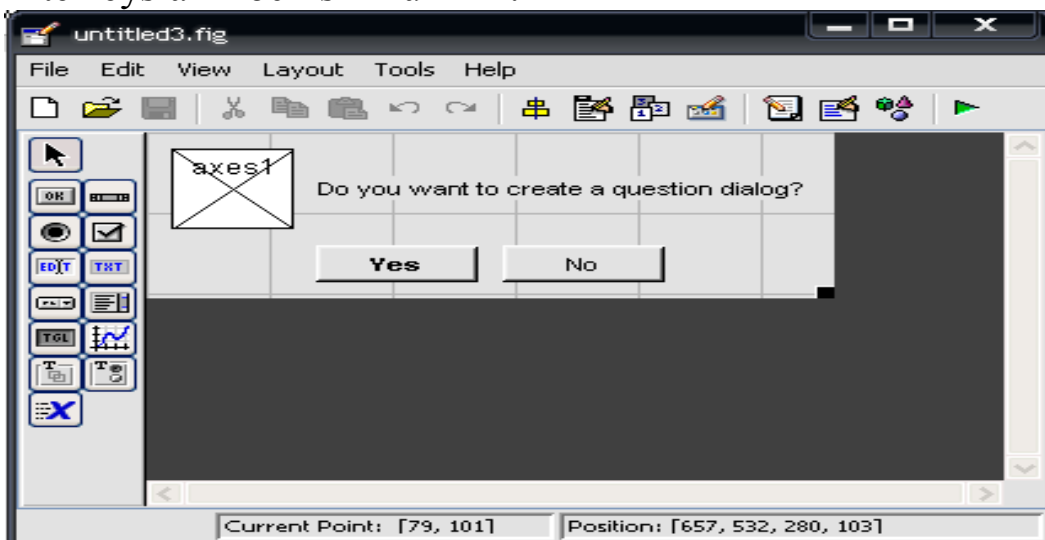
13.9 – rasm. GUI with Uicontrols oynasi.

3. GUI with Axes and Menu. Ushbu bo'lim ham 2-bo'lim kabi bir qancha aktiv ob'yektlarni o'z ichiga oladi. Bularga grafik chizish ob'yekti va menyu ob'yektlarni olish mumkin.



13.10 - rasm. Grafik komandasi aktiv.

4. Modal Question Dialog. Ushbu bo'limda bir nechta muloqot oynalar bilan ishlash jarayoni keltirilgan. Matlabda bir necha o'nlab muloqot oynalari mavjud. Bularga xatoliklarni bosmaga chiqaruvchi muloqot oynasi, xujjatlarni saqlash muloqot oynasi, saqlangan xujjatlarni ochish muloqot oynasi, ogohlantirish muloqot oynasi va boshqalar. Dasturda nafaqat yangi interfeys yaratibgina qolmasdan, oldin mavjud bo'lgan interfeyslarni ochish mumkin.



13.11 - rasm. Muloqot oynasi.

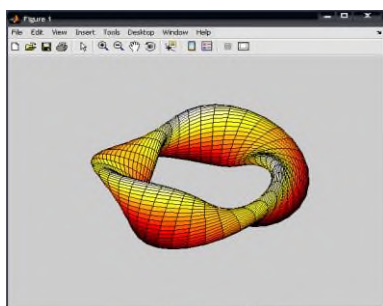
13.6. Uch o'lchovli grafiklar galariyasi

Matlabda 3-o'lchovli grafik imkoniyatlari bilan tanishish uchun professional tarzda bajarilgan maxsus grafik dasturlar galereyasi mavjuddir. Galereyaga demonstratsiya rejimidan ham (komanda Examples and Demos → komandalar oynasining help menyusi), va komandalar rejimidan ham (ma'lum faylni nomini terib), murojaat qilib kirish mumkin.

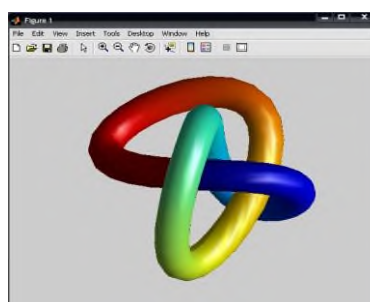
Galereya quyidagi shakl va fayllar bilan aniqlanadi:

Figura nomi	Fayl	Figura tuzilishi
Knot	Knot.m	Bo'langan (uzel) xalqa
Quiver	Quiv demo.m	Vektor hajmli maydon
Kleinll	Kleinl.m	Hajmli xalqa
Sruller	Sruller.m	Mebiusning hajmlisi
Hoops	Tory4.m	4 ta hajmli xalqalar
Slosh	Spharm2.m	O'rdakka o'xshash xalqani qurish
Modes	Modes.m	Uch o'lchovli sirt animasiya fazalarni ko'rsatish
Logo	Logo.m	MATLAB sistemasi logotipini qurish

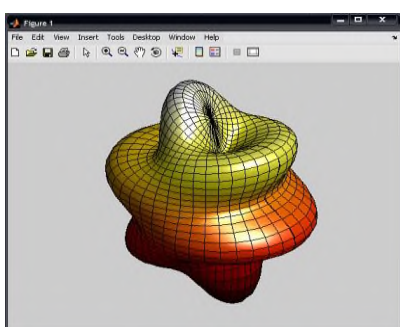
Misollar:



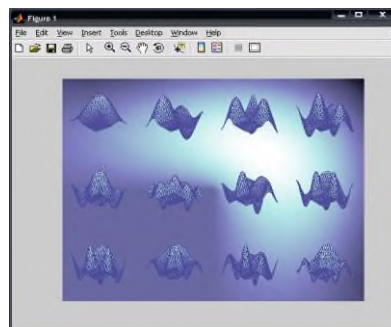
knot.m



cruller.m



spharm2.m



modes.m

13.12 - rasm. Murakkab figuralar grafiklari.

Nazorat savollari

1. Animatsiyani qanday buyruqlar yordamida hosil qilinadi?
2. Nuqtani fazoda harakatlantirishga misollar keltiring.
3. Dtskriptorli grafika nima?
4. Foydalanuvchi interfeysini qanday usullarda yaratish mumkin?
5. Foydalanuvchi intarfeysini yaratish usullari orasidagi farqlarni tushuntirib bering.
6. Uch o'lchovli grafik galereyasiga qanday shakllar kiradi ?
7. Uch o'lchovli grafikga misollar keltiring.

Mustaqil ishlash uchun misollar

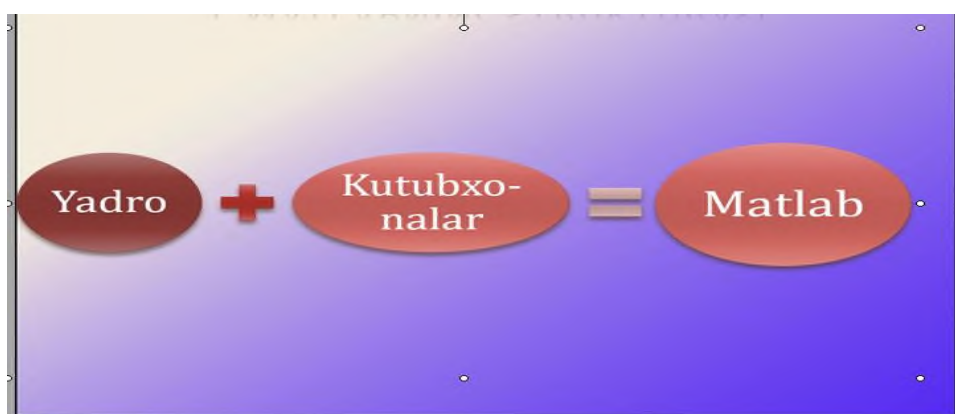
1. $y=\sin x$ va $y=x^2\cos x$, $x\in[-4\pi; 4\pi]$, funksiyalar grafigini bir grafik oynada har xil rangda hosil qiling.
2. Radiuslari 20 va 30 ga teng silindr grafiklarini bir-biriga urunadigan holatda chizing.
3. 2 ta sfera ustiga 1 ta silindrni gorizontal holatda joylashtiring.
4. Line komandasidan foydalanib, dekart tekisligining 7 ta nuqtasidan o'tuvchi chiziqni '0' belgi bilan markerlab havo rangda chiqaring.
5. 60° burchak ostida kesishuvchi radiuslari 12 va 11 ga teng silindrlarni grafigini chizing.
6. $Z=x^2+y^2$, $z=x^3+y^3$, $x,y\in[-7; 7]$ sirtlarni grafiklarini hosil qiling.
7. Radiusi 0.8 ga teng vertikal sfera ustiga sfera joylashtiring.
8. 4, 5 va 6 misolda chizilgan 4 ta grafikni grafik oynani 4 ga bo'lib joylashtiring.
9. Radiusi 5 ga teng bo'lgan sferani asoslari $z=x+2y+1$ tekislik sirtiga parallel holda joylashtiring.
10. 4 ta har xil radiusli silindrlarni chapdan o'ngga radiuslari oshib borishi tartibida joylashtiring.

14. MATLAB PAKETINING KENGAYTMASI, BIBLIOTEKALAR

14.1. MATLAB strukturasi

Matlab strukturasi (tuzilishini) umuman olganda ikkita katta qismdan iborat deb hisoblash mumkin: yadro va bibliotekalar. Matlabning yadrosi asosan umumiy xarakterga ega bo'lgan operatsiyalar va funksiyalardan iboratdir. Bibliotekalar esa tor mutaxassislikdagi funksiyalardan iborat bo'lib, foydalanuvchilar uchun shu mutaxassisliklar doirasida ma'lumotlarga ishlov berish va hisoblashlarni bajarish imkoniyatini beradi. Matlab tizimida juda ko'p bibliotekalar mavjud bo'lib, ularni bir qismi Math.Works kompaniyasi tomonidan yaratilgan, bir qismi esa foydalanuvchilar tomonidan yaratilgandir.

Bibliotekalarni yana kengaytirish imkoniyati ham mavjud.



14.1 - rasm. Matlabning tuzilishi.

Bibliotekalardagi funksiyalar matematik logika, boshqarish nazariyasi, neyron to'rlari, matematik modellashtirish, signallarga ishlov berish va boshqa yo'nalishlardagi masalalarni echishga mo'ljallangan. Matlabning standart bibliotekalaridan bir nechtasini ko'rib chiqamiz.

Wavelet bibliotekasi – Matlab bazasida shakllantirilgan funksiyalar to'plami bo'lib, ular elementar to'lqinlar va elementar to'lqinlar paketini ishlatuvchi signal va tasvirlarni Matlab strukturasi chegarasida analiz va sintez qilishning instrumental vositalari bilan ta'minlab beradi. Instrumental vositalar ikki xil bo'lishi mumkin:

- komandalar qatori funksiyalari;
- grafik interaktiv instrumental vositalar.

Birinchi turdagi vositalar- shunday funksiyalarki, ularni bevosita komandalar qatoridan chaqirish mumkin bo'ladi. Bu funksiyalar asosan m-fayllar bo'lib, ular elementar to'lqinlarning maxsus analizini amalga oshiradi. Bu funksiyalar kodini **type <funksiya nomi>** komandasi yordamida o'rganish mumkin .

Funksiya bosh qismini (yordamchi qism)

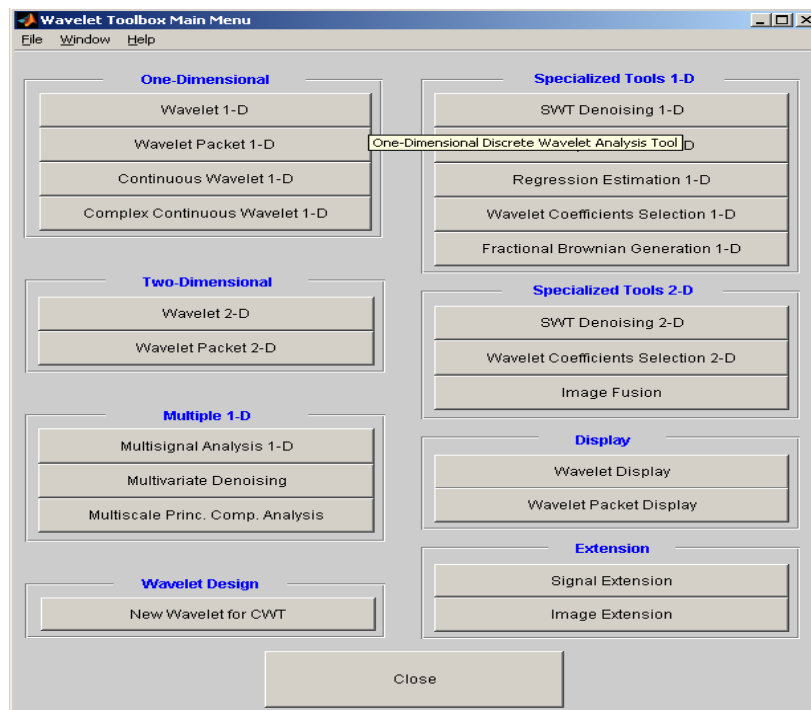
help <funksiya nomi>

komanda orqali ko'rish mumkin bo'ladi. Wavelet bibliotekasining barcha funksiyalari ro'yxatini

help wavelet

komandasi ko'rsatib beradi. Bibliotekadagi ixtiyoriy funksiyani ishlatishni o'zgartirish mumkin. Buning uchun uni nusxasini nomi o'zgartirilgan m-faylga joylashtiriladi va kerakli o'zgartirishlar amalga oshiriladi. Wavelet bibliotekasini yangi funksiyalar bilan kengaytirish imkoniyati ham mavjuddir.

Ikkinchi turdagi instrumental vositalarga grafik instrumental vositalar interfeysi majmuasi kiradi. Bu vositalar yordamida keng funksional imkoniyatlarga ega bo'lish mumkin. Bu vositalarga komandalar qatoridan **wavemenu** komandasi orqali murojaat qilinadi va quyidagi oyna ko'rinishi chiqadi:



14.2-rasm. Wavemenu ning natijasi.

14.2. Image Processing bibliotekasi

Bu biblioteka shunday funksiyalar majmuasiki, ular Matlabning imkoniyatlarini yanada kengaytiradi va ular yordamida tasvirlarga ishlov berish bo'yicha keng diapazondagi amaliyotlarni bajarish mumkin bo'ladi. Ulardan:

- geometrik amaliyotlar;
- chiziqli filtrlar va filtrlarni ishlab chiqish;
- almashtirishlar;
- tasvirlarni analiz qilish;
- ikkilik tasvirlar bilan amaliyotlar.

Bu bibliotekaning II versiyasi I ga nisbatan ancha ko'p afzalliklarga ega : bibliotekaning II versiyasida I versiyaning ko'p funksiyalari tezlik va kam xotira ishlatish maqsadida ko'chirib yozilgandir va boshqa yangi funksiyalar ham kiritilganki, ular bibliotekaning imkoniyatlarini yanada kengaytiradi. Bibliotekaning barcha funksiyalari ro'yxatini olish uchun

help win images/contents

komandasidan foydalaniladi.

14.3. Signal Processing bibliotekasi

Signal Processing bibliotekasi – Matlab bazasida shakllantirilgan instrumental vositalar to'plami bo'lib, signallarga ishlov berish bo'yicha keng qamrovli (diapazondagi) operatsiyalarni amalga oshiradi. Bunday amaliyotlarga to'lqinlarni o'zgartirishdan tortib, parametrik modellashtirishdagi va spektral analizdagi filtrlarni ishlab chiqish va amalga oshirishlar kiradi.

Biblioteka ikkita kategoriyadagi instrumentlar vositasidan iborat:

- signallarga ishlov beruvchi funksiyalar;
- grafik interaktiv instrumental vositalar.

Birinchi kategoriyadagi instrumentlar vositasi shunday funksiyalardan tuzilganki, ularni komandalar qatoridan yoki boshqa ilovalardan chaqirish mumkin bo'ladi.

Ikkinchi kategoriya – bu shunday interaktiv instrumental vositalarki, ular yordamida foydalanuvchining grafik interfeysi (GUI) orqali ko'p funksiyalarga murojaat qilish mumkin.

GUI ga asoslangan instrumental vositalar filtrlarni loyihalash, analiz qilish va bajarish uchun integrallangan muhit yaratib beradi. Masalan, GUI yordamida:

- filtr xarakteristikasini grafik jihatdan tahrirlash uchun “sichqoncha” dan foydalanish; yoki signal oq’maligini vizual ekran lineykasi yordamida o’lchash;
- menyu pozitsiyasidan yoki klavishlardan foydalanib, signalni ovoz apparatlari vositasida bajarilishi (proigrat);
- ochilayotgan menyudan foydalanib , signalning parametrlarini va hisoblash usullarini sozlash mumkin bo’ladi.

14.4. Simulink va Stateflow paketi

Simulink paketi - dinamik tizimlarni modellashtirish va simulyatsiya qilish uchun fanda va ishlab chiqarishda ko’p qo’llaniladigan dasturlar paketi hisoblanadi. Simulinkdan foydalanib, namunalar yordamida yangi modellar tuzish hamda mavjud modellarga komponentalar qo’shish mumkin bo’ladi. Simulyatsiya interaktiv bo’lgani uchun, ish jarayonida parametrlarni o’zgartirib, uning natijasini darrov ko’rsa bo’ladi. Matlabning barcha instrumental vositalariga to’g’ridan-to’g’ri kirish imkoniyati mavjud bo’lgani uchun, natijalarni olish, ularni analiz qilish va kerakli grafiklarni qurish mumkin .

Simulinkdan foydalanib real obyektlarning chiziqli bo’lmagan modellarini qurish va o’rganish mumkin. Simulink paketi uzluksiz vaqt jarayonida modellashtirilgan chiziqli va chiziqli bo’lmagan tizimlarni berilgan vaqt oraliq’ida qo’llab turadi. Modellashtirishda Simulink modelni blok-sxema sifatida yaratish uchun, foydalanuvchining grafik interfeysi bilan ta’minlab beradi. Bunda sichqoncha bilan bajariladigan «**click-and-drag**» dan foydalaniladi. Bu interfeys yordamida modelni xuddi qalam-qoq’oz ishlatgandek chizish mumkin bo’ladi. Bunday imkoniyat avvalgi paketlarda mavjud bo’lmagan. Undan tashqari, Simulink har xil bloklar (qabul qiluvchilar, manbalar, chiziqli va chiziqli bo’lmagan komponentalar, birlashtiruvchilar) dan iborat bo’lgan bibliotekani ulaydi.

Model aniqlangandan keyin uni yoki integrallash metodlaridan yoki Simulink menyusidan yoki komandalar oynasida Matlab komandalaridan foydalanib, bajarilishga (simulyatsiya) qo’yish mumkin. Interaktiv ishlash uchun menyu qulay bo’lsa, paketli modellashtirishni bajarishda komandalar oynasi qulay bo’ladi. Maxsus namoyish bloklaridan foydalanib, simulyatsiya bajarilmasdan avval simulyatsiya natijalarini ko’rish mumkin. Modellashtirish natijalarini Matlabning ishchi fazosiga joylab qo’yish mumkin.

Endi **Stateflow** dasturining imkoniyatlari bilan tanishib chiqamiz. **Stateflow**-boshqarish va nazorat qilishning murakkab masalalarini loyihalashtirish va rivojlantirish uchun kuchli grafik instrument hisoblanadi. Stateflowdan foydalanib:

-chekli avtomatlar nazariyasiga asoslangan kompleks reaktiv tizimlarni vizual modellashtirish va simulyasiya qilish;

-determinirlangan markaziy boshqaruv tizimlarini loyihalashtirish va rivojlantirish;

-blok-sxemalarda va Stateflowning bitta diagrammasidagi holatlar o'zgarishida belgilashlar tizimidan foydalanish;

-loyihalarni oson o'zgartirish, natijalarni baholash va loyihaning ixtiyoriy bosqichida tizimning o'zini tutishini tekshirish;

- Matlab va Simulink bilan integrallashganlik afzalligidan foydalanish;

Blok-sxemalardagi belgilashlar tizimi - dasturning umumiy strukturasi xuddi sikl operatori for va shartli operator if –end kabi effektiv usulda berish imkonini yaratish mumkin.

Stateflow paketi imkoniyatlaridan quyidagilarda foydalanilgan:

1) Joriy qilingan tizimlar:

-aviatsiya (samolyotlar);

-avtomobil sanoati;

-berilganlarni uzatish;

-dasturlanuvchi mantiqiy nazoratchilar;

-tijorat;

2) Inson-mashina interfeysi:

-foydalanuvchining grafik interfeysi;

3) Gibrid tizimlar:

-havo yo'llarini boshqarish tizimi.

Stateflow quyidagi komponentalardan tashkil topgan:

- ♦ Stateflowning grafik redaktori;
- ♦ Stateflowning yo'l boshlovchisi;
- ♦ Stateflowning qidiruv vositalari;
- ♦ Stateflow modellashtirish obyekt kodini generatori;
- ♦ Stateflow sozlagichi.

Nazorat savollari

1. Matlab tizimi strukturasi qanday bo'limlardan iborat?
2. Wavelet bibliotekasi qanday funksiyalardan iborat.

3. Wavelet bibliotekasi funksiyalar ro'yxatini ko'rish komandasi qanday?

4. Wave menu – komandasi qanday vazifani bajaradi?

5. Image Processing bibliotekasining vazifalari.

6. Help win images/contents – komandasini tushuntirib bering.

7. Signal Processing bibliotekasi tuzilishi qanday?

8. Simulink paketi qanday dastur?

9. Simulink ning imkoniyatlari.

10. Stateflow dan foydalanadigan tizimlar.

11. Stateflow nima?

12. Stateflow qanday komponentlarga ega?

15. SIMULINK PAKETI–DINAMIK TIZIMLARNI VIZUAL MODELLASHTIRISH TIZIMI

15.1. Simulink paketining umumiy vazifalari

Oxirgi yillarda Simulink paketi fan va sanoatda dinamik sistemalarni modellashtirish va simulyatsiya qilishda eng keng foydalaniladigan dasturiy paketlardan biridir.

Simulink ni ishlatib, namunalardan osongina model yaratish mumkin yoki mavjud modellarga komponentlar qo'shish mumkin. Simulyatsiya qilish jarayoni interaktiv bo'lgani uchun, ish jarayonida parametr qiymatlarini o'zgartirib, natijalarni o'zgarishini ko'rish mumkin bo'ladi. Bu paketdan Matlabning barcha analiz qiluvchi instrumental vositalariga to'q'ridan-to'q'ri kirish imkoniyati bor. Shuning uchun natijalarni analiz qilish va kerakli grafiklarni qurish va o'rganish mumkin.

Simulink yordamida real chiziqsiz bo'lgan modellarni o'rganish va qurish mumkin. Bunday modellar, bizga ma'lumki, qarshilik, ishqalanish, havo qarshiligi mexanizmlarni sirpanishi va boshqalarni hisobga olish imkoniyatini beradi.

Simulink – bu dinamik sistemalarni modellashtirish, simulyat siya va analiz qilishga mo'ljallangan dasturlar paketidir. Bu paket uzluksiz vaqt mobaynida modellashtirilgan chizikli va chiziqsiz bo'lgan, ma'lum vaqt oraliq'ida berilgan tizimlarni qo'llab-quvvatlaydi. Sistemalar har xil tezlikda bo'lishi, yani sistemani har xil bo'limi har xil tezliklarda bajarilishi mumkin.

15.2. Modellashtirishda Simulink paketining roli

Modellashtirish uchun Simulink paketi modelni blok-sxema sifatida shakllantirish uchun foydalanuvchining grafik interfeysi bilan ta'minlaydi. Bunda “sichqoncha” vositasida “click-and-drag” operatsiyasidan foydalaniladi. Bu interfeys yordamida modellarni qalam va qoq'oz ishlatib “chizish” mumkin. Simulink har xil bloklardan (qabul qiluvchi, manbalar, chizikli va chiziqsiz komponentalar, ulagich (soediniteli) lar) dan iborat bo'lgan bibliotekani ulab beradi. Bundan tashqari, foydalanuvchi o'z bloklarini ishlab chiqishi va sozlashi mumkin.

Barcha modellar iyerarxik tuzilishga ega. Shuning uchun, modellarni o'suvchi yoki kamayuvchi nuqtai-nazaridan kelib chiqib shakllantirish mumkin. Sistemani yuqori darajada (uroven) qarash mumkin va bloklarda

ikkita (“шелчок”) “bosish” natijasida darajalar (уровни) bo’yicha pasayib, model detallarining o’suvchi darajalariga kirishni ta’minlash mumkin bo’ladi. Bu nuqtai-nazar (подход) yordamida modelning tuzilishini va uning qismlarini qanday birgalikda ishlashini tushunishni ta’minlab beradi.

Model aniqlangandan keyin uni foydalanish uchun qo’ysa bo’ladi. Bunday ishni integrallash metodidan, yoki Simulink menyusidan, yoki MATLAB komandalar oynasidan ma’lum komanda kiritib amalga oshirish mumkin. Interaktiv ishlash jarayonida menyudan foydalanish qulay bo’lsa, paketli modellashtirish jarayonida komandalar oynasidan foydalanish qulaydir. Maxsus demonstratsion bloklardan foydalanib, simulyatsiya bajarilmayotgan bo’lsa ham, simulyatsiya natijalarini ko’rish mumkin. Bundan tashqari parametrlarni o’zgartirib, birdaniga u qanday natija berganini ko’rish mumkin. Modellashtirish (simulyatsiya) natijalarni Matlabning ishchi fazosiga joylashtirib, keyinchalik qayta ishlab vizualizatsiya qilish mumkin bo’ladi.

Modellarni analiz qilish instrumentlariga chiziqshtirish va qurish (podstroyka) vositalari kiradi. Bu vositalar komandalar oynasidan chaqiriladi. Undan tashqari Matlabning ko’p instrumental vositalari va bibliotekalaridan ham foydalanish mumkin. Matlab va Simulink tizimlarining hisobiga bu tizimning ixtiyoriy nuqtasida modellashtirish, analiz qilish va modellarni to’g’rilash mumkin bo’ladi.

15.3.Stateflow programmasi

Stateflow kuchli grafik instrument bo’lib, boshqarish va kontrol qilishning murakkab masalalarini loyihalashtirish va rivojlantirish uchun mo’ljallangan.

Stateflow dan foydalanib:

- chekli avtomatlar nazariyasiga asoslangan kompleks reaktiv sistemalarni vizual modellashtirish va simulyatsiya qilish;
- markaziy kontrolning determinirlangan sistemalarini loyihalashtirish va rivojlantirish;
- blok-sxemalarda belgilashlar sistemasini ishlatish, State flow paketi bitta diagrammasidagi holat o’zgarishlari belgilashlar sistemasini ishlatish;
- loyihaaning ixtiyoriy stadiyasida loyihasini oson o’zgartirish, natijalarni baholash va tizimni o’zini tutishini bilish;
- Matlabning Simulink bilan integrallashganlik afzalligidan foydalanish;

kabi ishlarni amalga oshirish mumkin.

State flow quyidagi komponentalardan iborat:

- State flow ning grafik taxriri;
- State flow ning provodnigi (belgilovchisi);
- State flow ning qidiruv vositalari;
- State flow modellashtirish ob'ektlashgan kodining generatori;
- State flow sozlagichi;

State flow paketini qo'llanilishi quyidagi yo'nalishlarda bo'lishi mumkin (tadbiq etilgan sistemalar):

- aviatsiya (samolyotsozlik);
- avtomobil sanoati;
- ma'lumotlarni uzatish;
- kommertsiya (tijorat);

Inson-mashina interfeysi:

- foydalanuvchining grafik interfeysi.

Gibrid sistemalar:

- havo harakatini boshqarish sistemasi;

Stateflow paketi, chekli avtomatlar nazariyasini ishlatib, murakkab sistemalar faoliyatini aniq va qisqa qilib ochib beradi. Undan tashqari, bu paket sistema va uning loyihasiga qo'yiladigan texnik talablarni bir-biriga yaqinlashtiradi. Bu juda sodda amalga oshiriladi: loyiha yaratiladi va ssenariyning har xil variantlari ko'riladi, integratsiyalar esa State flow paketi diagrammasi modelning kerakli faoliyatini hosil qilmaguncha davom etadi.

Nazorat savollari

1. Simulink dasturining mohiyati nima?
2. Simulink qanday bloklardan iborat bo'lishi mumkin?
3. Simulink da modellar qanday tuzilishga ega?
4. Stateflow dasturining mohiyati nima?
5. Stateflow paketi qanday yo'nalishlarda effektiv qo'llanishi mumkin?

$A^*=[A,b]$ -kengaytirilgan matritsani kiritamiz. Chiziqli algebra kursidan ma'lumki (Kroneker-Kapelli teoremasi), A va A^* matritsalarining ranglari teng bo'lsa, (1) yoki (2) sistemaning yechimi mavjud bo'ladi.

16.2. Chiziqli tenglamalar sistemasini yechish usullari

Chiziqli tenglamalar sistemasini yechishning aniq usullaridan keng qo'llaniladiganlari Gauss, Kramer va teskari matritsa usullaridir, taqribiy usullarga esa iteratsiyalar (ketma-ket yaqinlashish), Zeydel va kichik kvadratlar usullarini keltirish mumkin.

Aniq usullardan **Kramer usulini** ko'rib chiqamiz:

Buning uchun $\det(A) \neq 0$ bo'lishi kerak. Usulni to'liq keltirish uchun sistemaning asosiy matritsasi A ning k -ustun elementlarini ozod had b bilan almashtirib, A_k , $k=1, n$, matritsalar hosil qilamiz. U holda $\det(A) \neq 0$ shart asosida yechimni topish uchun

$$x_k = \frac{\det(A_k)}{\det(A)}, \quad k=1, 2, \dots, n$$

tengliklardan foydalanish mumkin. Bu yerda foydalanilgan $\det(A)$ Matlab funksiyasi bo'lib, A matritsaning determinantini hisoblab beradi.

Taqribiy usullardan **iteratsiya usulini** keltiramiz. Buning uchun (1) sistemani quyidagi ko'rinishga keltiramiz:

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \dots + \alpha_{1n}x_n, \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + \dots + \alpha_{2n}x_n, \\ \dots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \dots + \alpha_{nn-1}x_{n-1} \end{cases} \quad (3)$$

Bu erda

$$\beta_i = \frac{b_i}{a_{ii}}, \quad \alpha_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad i \neq j, \\ \alpha_{ij} = 0, \quad i = j, \quad i, j = 1, 2, \dots, n.$$

U holda

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix}$$

belgilashlar kiritib, (3) ni quyidagicha yozib olamiz:

$$x = \beta + \alpha x \quad (4)$$

Endi (4) sistemani ketma-ket yaqinlashish (iteratsiya) usuli bilan yechamiz. Boshlang'ich yaqinlashish uchun $x^{(0)} = \beta$ ozod hadni olamiz va ketma-ket keyingi yaqinlashishlarni hosil qilamiz:

$$x^{(1)} = \beta + \alpha x^{(0)};$$

$$x^{(2)} = \beta + \alpha x^{(1)};$$

...

$$x^{(k+1)} = \beta + \alpha x^{(k)}; \dots$$

Agar $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$ sonlar ketma-ketligi chekli limitga ega bo'lsa, u holda bu limit (3) yoki (4) sistemaning yechimi bo'ladi. Yaqinlashishlarni ochiq holda quyidagicha yozish mumkin:

$$x_i^{(0)} = \beta_i, \\ x_i^{(k+1)} = \beta_i + \sum_{\substack{j=1 \\ j \neq i}}^n \alpha_{ij} x_j^{(k)}, \quad i = \overline{1, n}, \quad k = 0, 1, 2, \dots \quad (5)$$

Yechimni taqribiy hisoblashning ana shunday usuli iteratsiya usuli deyiladi. Iteratsiya protsessining yaqinlashuvchi bo'lishining yetarli shartini quyidagi teoremda keltiramiz:

Teorema. Agar o'zgartirilgan (3) sistemada quyidagi shartlardan

$$1) \quad \sum_{j=1}^n |\alpha_{ij}| < 1, \quad i = 1, 2, \dots, n.$$

$$2) \quad \sum_{i=1}^n |\alpha_{ij}| < 1, \quad j = 1, 2, \dots, n.$$

biri bajarilsa, u holda, ixtiyoriy boshlang'ich nuqta $x^{(0)}$ uchun hosil qilingan (5) iteratsiya jarayoni yagona yechimga yaqinlashuvchi bo'ladi.

Vektor ko'rinishidagi (2) sistemani $\det A \neq 0$ bo'lgan holda teorema shartini qanoatlantiradigan ekvivalent sistemaga keltirish mumkin:

$$(A^{-1} - \varepsilon)Ax = Db, \quad D = A^{-1} - \varepsilon; \quad (6)$$

bu yerda $\varepsilon = [\varepsilon_{ij}]$ - yetarli kichik sonlardan iborat bo'lgan matritsa. Yuqoridagi (6) sistemada qavsni ochib, $\alpha = \varepsilon A$, $\beta = Db$ belgilashlardan foydalanib iteratsiya usulini qo'llash uchun qulay bo'lgan (4) ko'rinishdagi sistemani olamiz:

$$x = \beta + \alpha x,$$

Yuqorida keltirilgan $\varepsilon = [\varepsilon_{ij}]$ matritsada ε_{ij} elementlarni yetarli kichik qilib olinsa, teorema shartlari bajariladi.

16.3. Chiziqli tenglamalar sistemasini echishda MATLAB usullari

Chiziqli tenglamalar sistemasini yechish uchun Matlab funksiyalari (usullari) juda ko'p bo'lib, biz ulardan bir nechtasini keltiramiz.

1) $x = A \setminus B$ - "o'ngdan bo'lish" usuli ;

2) $\mathbf{x}=\mathbf{lsqnonneg}(\mathbf{A},\mathbf{B})$ - $Ax=B$ chiziqli tenglamalar sistemasini kichik kvadratlar usuli bilan yechadi. Bunda A -($n \times n$) o'lchovli, B -($n \times 1$) o'lchovli, $x_i \geq 0, i=1,2,\dots,n$. Minimallashtirish kriteriyasi: $B-Ax$ ning ikkinchi normasini minimallashtirish;

3) $\mathbf{x}=\mathbf{lsqnonneg}(\mathbf{A},\mathbf{B},\mathbf{x0})$ - iteratsiyalar uchun chiziqli tenglamalar sistemasining aniq berilgan nomanfiy boshlanq'ich qiymatlarda yechib beradi;

4) $[\mathbf{x},\mathbf{w}]=\mathbf{lsqnonneg}(\dots)$ - echim bilan birga qoldiqlar vektori kvadrati ikkinchi normasini qaytaradi;

5) $[\mathbf{x},\mathbf{w},\mathbf{w1}]=\mathbf{lsqnonneg}(\dots)$ - xuddi avvalgi buyruq kabi, yana qoldiqlar vektori $w1$ ni qaytaradi;

6) $\mathbf{bicg}(\mathbf{A},\mathbf{B})$ - $Ax=B$ tenglamaning x yechimini qaytaradi; A ($n \times n$), B ($n \times 1$). Bunda hisoblash iteratsiyalar yaqinlashguncha yoki $\min\{20,n\}$ gacha bajariladi;

7) $\mathbf{bicg}(\mathbf{A},\mathbf{B},\mathbf{tol})$ - echimni tol xatolik bilan qaytaradi;

8) $\mathbf{bicg}(\mathbf{A},\mathbf{B},\mathbf{tol},\mathbf{maxit})$ - avvalgi buyruq kabi, yechimni undan tashqari maxit-maksimal iteratsiyalar soni bilan qaytaradi.

16.4. Chiziqli tenglamalar sistemasini yechishga doir misollar

1. Tenglamalar sistemasini o'ngdan bo'lish, iteratsiyalar va Kramer usulida yeching, topilgan yechimlarni solishtiring.

$$\begin{cases} 2x + y + z + t = 8 \\ 3x - y - 2z + t = 2 \\ x + 2y - 3z + 2t = 8 \\ 5x - 2y + 3z - t = 1 \end{cases}$$

Yechimni topish uchun komandalar oynasidan foydalanamiz.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> a= [2 1 1 1; 3 -1 -2 1; 1 2 -3 2; 5 -2 3 -1];b=[8;2;8;1];
>> a= [8 1 1 1; 2 -1 -2 1; 8 2 -3 2; 1 -2 3 -1];
>> a2= [2 8 1 1; 3 2 -2 1; 1 8 -3 2; 5 1 3 -1];
>> a1= [8 1 1 1; 2 -1 -2 1; 8 2 -3 2; 1 -2 3 -1];
>> a= [2 1 1 1; 3 -1 -2 1; 1 2 -3 2; 5 -2 3 -1];
>> a3= [2 1 8 1; 3 -1 2 1; 1 2 8 2; 5 -2 1 -1];
>> a4= [2 1 1 8; 3 -1 -2 2; 1 2 -3 8; 5 -2 3 1];
>> x=det(a1)/det(a);
>> y=det(a2)/det(a);
>> z=det(a3)/det(a);
>> t=det(a4)/det(a);
>> %Kramer uculida quyidagi echimni olamiz
>> X=[x y z t]

X =

     1     2     1     3

>> X=a\b

X =

     1.0000
     2.0000
     1.0000
     3.0000

>> %yuqorida o'ngdan bo'lish uculida olingan echim uctun ko'rinishda
>> |

```

16.1-rasm. Sistemaning yechimlari.

Endi xuddi shu tenglamalar sistemasini iteratsiya usuli bilan yechamiz va natijalarni solishtiramiz.

Yechimni iteratsiyalar usulida topish uchun quyidagi fayl-funksiyani tuzamiz:


```

Editor - C:\Documents and Settings\Kamil\Мои документы\MATLAB\iter.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Stack:
- 1.0 + 1.1 x % % % %
1 function zz=iter(a,b,x0,eps,n)
2 r1=1;
3 while r1
4 r1=0;
5 for i=1:n
6 x(i)=b(i)/a(i,i);
7 for j=1:n
8 if j~=i
9 x(i)=x(i)-a(i,j)*x0(j);
10 end
11 end
12 r1=r1+(abs(x(i)-x0(i)))>=eps;
13 end
14 x0=x
15 end
16 zz=x0

```

16.2- rasm. Yechimni iteratsiya usulida topish.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> %yuqorida o'ngdan bo'lish usulida olingan echim uctun ko'rinishda
>> xx=iter(a,b,0.00001,4)

x0 =
-7.0000 -11.0000 -16.6667 -61.0000

x0 =
92.6667 46.6667 148.3333 62.0000

x0 =
-253.0000 -45.3333 -312.6667 -816.0000

x0 =
1.0e+003 *
1.1780 0.9477 1.9730 2.1113

x0 =
1.0e+003 *
-5.0280 -1.7013 -7.2987 -9.9147

```

16.3-rasm. Iteratsiya jarayoni.

```

Command Window
i New to MATLAB? Watch this Video, see Demos, or read Getting Started.

    -1.6925    -0.8540    -2.5516    -3.5208

x0 =

    1.0e+308 *

    0.6926    0.3495    1.0442    1.4409

x0 =

   -Inf   -Inf   -Inf   -Inf

x0 =

    Inf    NaN    Inf    NaN

zz =

    Inf    NaN    Inf    NaN

xx =

    Inf    NaN    Inf    NaN

>>

```

16.4-rasm. Iteratsiya jarayoni.

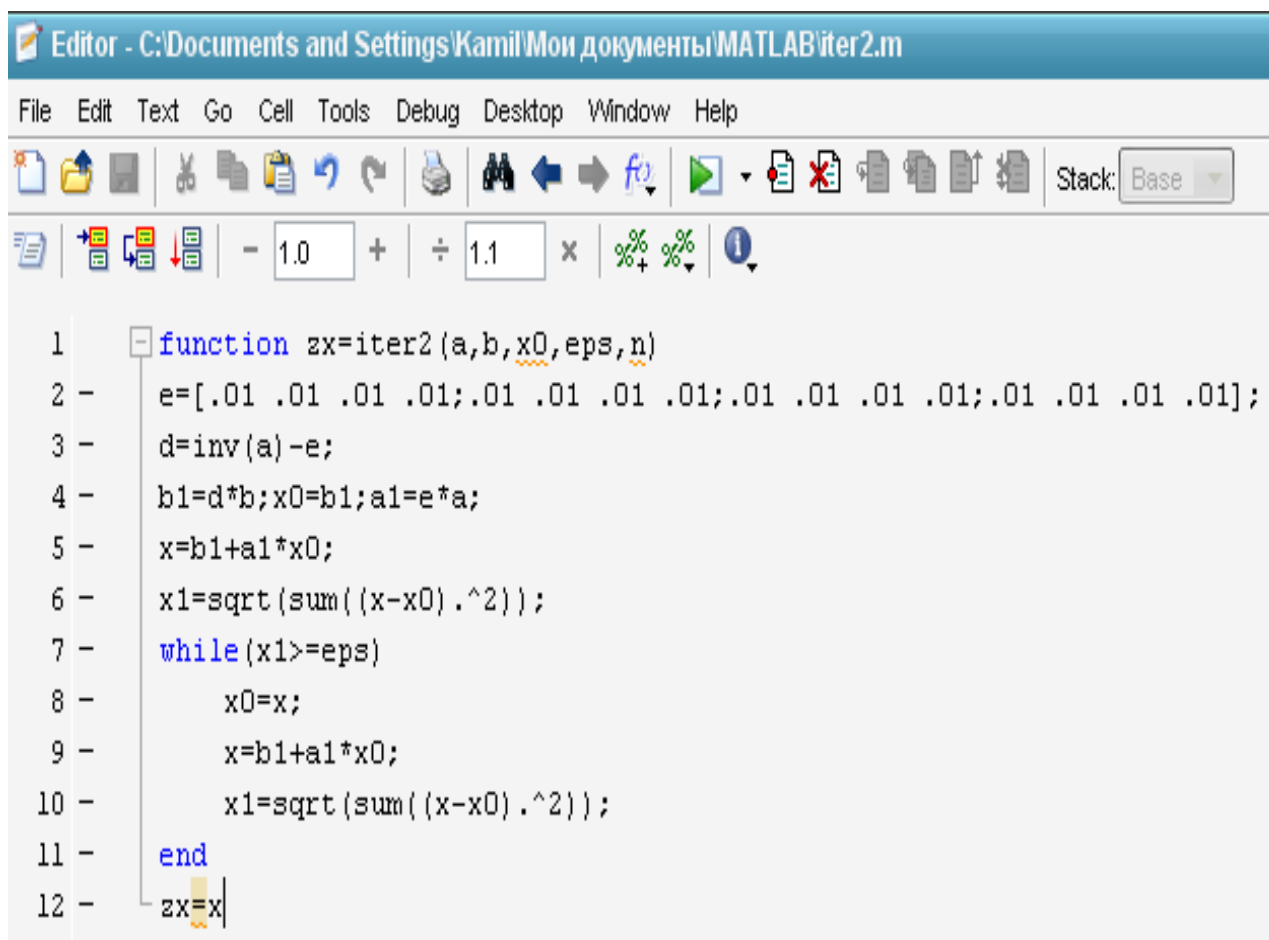
Natijalardan ko'rinib turibdiki, bu tenglamalar sistemasi yechimini topishga iteratsiyalar usulini to'g'ridan-to'g'ri qo'llaganimizda taqribiy yechimni aniqlash protsessi yaqinlashuvchi emas. Shuning uchun berilgan tenglamalar sistemasida quyidagicha o'zgartirishlar amalga oshiramiz:

```

e=[0.01 0.01 0.01 0.01; 0.01 0.01 0.01 0.01;
    0.01 0.01 0.01 0.01;0.01 0.01 0.01 0.01];
d=inv(a)-e; b1=d*b; a1=a*e; x0=b;

```

U holda hosil bo'lgan $x=b_1+a_1x$ tenglamalar sistemasi yuqorida keltirilgan teorema shartlarini qanoatlantiradi. Iteratsion algoritmi ishlashini yangi iter2 fayl-funksiya hosil qilib tekshiramiz.



```
Editor - C:\Documents and Settings\Kamil\Мои документы\MATLAB\iter2.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
- 1.0 + 1.1 x % % !
1 function zx=iter2(a,b,x0,eps,n)
2 e=[.01 .01 .01 .01;.01 .01 .01 .01;.01 .01 .01 .01;.01 .01 .01 .01];
3 d=inv(a)-e;
4 b1=d*b;x0=b1;a1=e*a;
5 x=b1+a1*x0;
6 x1=sqrt(sum((x-x0).^2));
7 while(x1>=eps)
8     x0=x;
9     x=b1+a1*x0;
10    x1=sqrt(sum((x-x0).^2));
11 end
12 zx=x
```

16.5-rasm. Yangi fayl-funksiya.

Hosil qilingan iter2 fayl-funksiyasiga argumentlar a, b, x_0, eps, n larning qiymatlarini komandalar oynasida hosil qilib, murojaat qilamiz va quyidagi natijalarni olamiz:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> a= [2 1 1 1; 3 -1 -2 1; 1 2 -3 2; 5 -2 3 -1];
>> b= [8;2;8;1];
>> iter2(a,b,b,0.0001,4)

zx =

    1.0000
    2.0000
    1.0000
    3.0000

ans =

    1.0000
    2.0000
    1.0000
    3.0000

>> %Topilgan echim qiymatlaridan ko'rinib turibdiki, iteratsiya usulida topilgan t
>> %taqribiy echim ham aniq echimga teng. Demak, bu holda taqribiy usulda ham aniq
>> %-lik juda katta.
```

16.6 –rasm. Iteratsiya usulida topilgan yechim.

Nazorat savollari

1. Chiziqli tenglamalar sistemasi va uning yechimi nima?
2. Chiziqli tenglamalar sistemasini qanday yechish usullari bor?
3. Chiziqli tenglamalar sistemasini yechishning qaysi taqribiy usullari mavjud?
4. Chiziqli tenglamalar sistemasini yechish uchun iteratsiyalar usuli.
5. Chiziqli tenglamalar sistemasini yechish uchun Matlab usullari.
6. Chiziqli tenglamalar sistemasini yechish uchun Kramer usuli.

Mustaqil ishlash uchun misollar

$$\begin{cases} x_1 - x_2 + x_3 = 5 \\ 2x_1 - x_2 + x_3 = -8 \\ x_1 + x_2 + 2x_3 = 21 \end{cases}$$

$$\begin{cases} x_1 - 4x_2 = -5 \\ x_1 + 4x_2 + x_3 = -1 \\ 2x_1 - 3x_2 + x_3 = -7 \end{cases}$$

$$\begin{cases} x_1 + 2x_2 - x_3 = 5 \\ 4x_1 + x_2 - 5x_3 = -3 \\ x_1 - 3x_2 + x_3 = -6 \end{cases}$$

$$\begin{cases} 2x_1 - 4x_2 + 3x_3 + 7x_4 = 4 \\ -2x_1 + x_2 - x_3 + 3x_4 = -3 \\ 4x_1 - 3x_2 + x_3 + 5x_4 = 0 \\ -x_1 + 2x_2 - x_3 + x_4 = -1 \end{cases}$$

$$\begin{cases} x_1 + 2x_2 - x_3 - 2x_4 = 3 \\ 3x_1 + 8x_2 - 4x_4 = 8 \\ 2x_1 + 2x_2 - 4x_3 - 3x_4 = -11 \\ 3x_1 + 8x_2 - x_3 - 6x_4 = 46 \end{cases}$$

$$\begin{cases} 2x_1 - 5x_2 + x_3 + 2x_4 = 3 \\ -3x_1 + 7x_2 - x_3 + 4x_4 = 5 \\ 5x_1 - 9x_2 + 2x_3 + 7x_4 = 2 \\ 4x_1 - 6x_2 + x_3 + 2x_4 = 17 \end{cases}$$

$$\begin{cases} 8x_1 + x_2 + 3x_3 = -1 \\ -4x_1 - 5x_2 + 2x_3 = 6 \\ 3x_1 + 2x_3 = 1 \end{cases}$$

$$\begin{cases} 3x_1 - 2x_2 + 6x_3 = -7 \\ 5x_1 + 8x_2 + x_3 = 2 \\ 2x_1 + x_2 - x_3 = -5 \end{cases}$$

$$\begin{cases} x_1 + 3x_2 - 5x_3 = 0 \\ x_1 + x_2 - x_3 = 0 \\ 3x_1 + x_3 = -2 \end{cases}$$

$$\begin{cases} x_2 - 2x_4 = 4 \\ 2x_1 - x_2 + x_3 = -1 \\ 3x_1 - 2x_2 + 4x_3 = -5 \\ 2x_1 + x_3 - 3x_4 = 2 \end{cases}$$

$$\begin{cases} 2x_1 + x_2 - x_3 + 4x_4 = 20 \\ 5x_1 + x_2 + 2x_3 - x_4 = 17 \\ -3x_1 + 2x_2 - x_3 + 2x_4 = 1 \\ x_1 + x_2 + 4x_3 - 2x_4 = -4 \end{cases}$$

$$\begin{cases} -3x_1 - 7x_2 - 8x_3 + 2x_4 = -4 \\ x_1 + 3x_2 + 4x_3 - 2x_4 = 2 \\ 2x_1 - x_2 + 3x_3 = 4 \\ 2x_1 + 4x_2 + 4x_3 = 3 \end{cases}$$

17. KUZATISH (STATISTICA) NATIJALARINI BIRLAMCHI QAYTA ISHLASH

17.1. Ma'lumotlarni statistik qayta ishlash masalasi

Umumiy holda boshlanq'ich ma'lumotlarni birlamchi qayta ishlash masalasi quyidagicha qo'yiladi: faraz qilaylik, tajribaviy o'rganish natijasida x miqdorning x_1, x_2, \dots, x_n qiymatlariga y miqdorning y_1, y_2, \dots, y_n qiymatlari mos qo'yilgan bo'lsin. Shu x va y miqdorni boq'lovchi $y=f(x)$ funksiyaning analitik ko'rinishini topish talab qilinadi. Bu funksiya berilgan x_1, x_2, \dots, x_n argument qiymatlarida mos ravishda y_1, y_2, \dots, y_n qiymatlarni qabul qilishi yoki shu qiymatlarga ma'lum aniqlikda yaqin bo'lishi shart. Mana shunday tajriba natijalarini bog'lovchi analitik funksiya $u=f(x)$ empirik deb ataladi. Bunday empirik bog'liqlikni aniqlashni ikkita bosqichga ajratish mumkin:

- parametrlarga bog'liq bo'lgan empirik formulani tanlash (strukturali identifikatsiya);
- tanlangan formuladagi parametrlarni aniqlash (parametrik identifikatsiya).

Strukturali identifikatsiya masalasi ancha murakkab masalalardan biri bo'lib, aniqlangan funksiya bir nechta analitik funksiyalar davomidan iborat bo'lishi mumkin. Funksiya ko'rinishi bir nechta parametrlarga bog'liq holda izlanadi. Ma'lum usullardan (masalan, kichik kvadratlar usuli) foydalanib, parametrlar aniqlanadi va aniqlangan funksiya qiymatlari berilgan x_i nuqtalarda hisoblanib, y_i qiymatlar bilan yaqinligi (ma'lum ma'noda) solishtiriladi. Yaqinlik qanoatlantirilsa, aniqlangan funksiya jarayonning modeli sifatida qabul qilinadi, aks holda empirik funksiya qurish yana boshqa ko'rinishdagi funksiya izlashdan boshlanadi.

17.2. Strukturali identifikatsiya

Funksiyani ko'rinishini aniqlash uchun bir yondoshuvni ko'ramiz. Bu yondoshuv berilgan ma'lumotlarning grafigidan foydalanishga asoslangandir. Shuning uchun berilgan ma'lumotlar grafigida katta sakrashlar ko'p bo'lganda bu yondoshuv yaxshi natija bermasligi mumkin.

Faraz qilaylik, qidirilayotgan funksiya $y=f(a,b,x)$ bir o'zgaruvchili va ikkita a hamda b parametrlarga ega bo'lsin. U holda empirik bog'liqlikni quyidagi funksiyalardan tanlab olish taklif etiladi:

- 1) Chiziqli funksiya $y=ax+b$;

- 2) Ko'rsatkichli funksiya $y=a*b^x$;
- 3) Kasr- ratsional funksiya $y= \frac{1}{ax+b}$;
- 4) Logarifmik funksiya $y=aln x+b$;
- 5) Darajali funksiya $y=ax^b$ (agar $b>0$ - bu parabolik boq'liqlik; agar $b<0$ - bu giperbolik boq'liqlik; agar $b=0$ - bu chiziqli boq'liqlik);
- 6) Giperbolik bog'liqlik $y=a+ \frac{b}{x}$;
- 7) Kasr-ratsional funksiya $y= \frac{x}{ax+b}$.

Empirik funksiyaning yuqoridagi funksiyalar ichidan tanlanishi bu bir yondoshuv bo'lib, umuman olganda bunday funksiyalar sinfi ixtiyoriy bo'lishi mumkin. Biz bu erda empirik bog'liqlikni tanlashni bir usulini ko'ramiz, xolos.

Bu usul bo'yicha, strukturali identifikatsiya qilishning boshlang'ich bosqichi bo'lib, ma'lumotlar massivlari x va y larning grafigini qurish hisoblanadi. Shundan so'ng, quyidagicha yordamchi hisoblashlarni bajaramiz:

x miqdorning qiymatlaridan yetarli darajada ishonchli bo'lgan va bir-biridan uzoqda joylashgan 2 ta nuqta olamiz, masalan, x_1, x_n lar bo'lsin. Bu nuqtalar uchun $x_{ar}=(x_1+x_n)/2$ - o'rta arifmetikni, $x_{geom}=\sqrt{x_1 * x_n}$ - o'rta geometrikni va $x_{garm} = 2(x_{geom})^2 / x_{ar}$ ni hisoblaymiz. Chizilgan grafik yordamida topilgan x miqdorlarning qiymatlariga mos bo'lgan y ning qiymatlarini aniqlaymiz:

$$x_{ar} \rightarrow y_1^*, \quad x_{geom} \rightarrow y_2^*, \quad x_{garm} \rightarrow y_3^* .$$

Yuqoridagi hisoblashlarni y miqdorning qiymatlari uchun ham bajaramiz:

$$y_{ar}=(y_1+y_n)/2, \quad y_{geom}=\sqrt{y_1 * y_n}, \quad y_{garm}=2*y_1*y_n/(y_1+y_n).$$

Hosil qilingan $y_{ar}, y_{geom}, y_{garm}, y_1, y_2, y_3$ sonlardan foydalanib, quyidagilarni hisoblaymiz:

$$\begin{aligned} \varepsilon_1 &= |y_1^* - y_{ar}|, & \varepsilon_2 &= |y_1^* - y_{geom}|, & \varepsilon_3 &= |y_1^* - y_{garm}|, \\ \varepsilon_4 &= |y_2^* - y_{ar}|, & \varepsilon_5 &= |y_2^* - y_{geom}|, & \varepsilon_6 &= |y_3^* - y_{ar}|, \\ \varepsilon_7 &= |y_3^* - y_{geom}|. \end{aligned}$$

Bu sonlarning minimumini aniqlaymiz: $\varepsilon = \min(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7)$. Minimal xatolik ε ni aniqlab, strukturali identifikatsiyani quyidagi qoida bo'yicha amalga oshiramiz:

1) agar $\varepsilon = \varepsilon_1$ bo'lsa, analitik bog'lanish chiziqli $y=ax+b$ ko'rinishda olinadi;

2) agar $\varepsilon = \varepsilon_2$ bo'lsa, analitik bog'lanish ko'rsatkichli $y=a*b^x$ ko'rinishda olinadi;

3) agar $\varepsilon = \varepsilon_3$ bo'lsa, analitik bog'lanish kasr-ratsional funksiya $y = \frac{1}{ax+b}$ ko'rinishda olinadi;

4) agar $\varepsilon = \varepsilon_4$ bo'lsa analitik bog'lanish logarifmik funksiya $y = a \ln x + b$ ko'rinishda olinadi;

5) agar $\varepsilon = \varepsilon_5$ bo'lsa analitik bog'lanish ko'rsatkichli funksiya $y = a * x^b$ ko'rinishda olinadi;

6) agar $\varepsilon = \varepsilon_6$ bo'lsa analitik bog'lanish giperbolik funksiya $y = a + \frac{b}{x}$ ko'rinishda olinadi;

7) agar $\varepsilon = \varepsilon_7$ bo'lsa analitik bog'lanish kasr-rasional funksiya $y = \frac{x}{ax+b}$ ko'rinishida olinadi.

Shunday qilib, ε qiymatiga mos ravishda aniq bir analitik formula (2 ta parametrli) tanlanadi. Analitik funksiya tanlashni yuqoridagidan farqli boshqa usulda ham amalga oshirsa bo'ladi.

17.3. Parametrik identifikatsiya

Empirik funksiyaning ko'rinishi topilgandan keyin a va b parametrlarning qiymati aniqlanadi.

Umuman olganda, parametrlarni aniqlashni bir nechta usullari mavjud. Biz ulardan

a) Tanlangan nuqtalar usuli;

b) Kichik kvadratlar usuli;

kabi usullarni ishlatamiz.

Tanlangan nuqtalar usuli eng sodda usul bo'lib, kam hisoblashlarni talab qiladi. Lekin bu usulning aniqligi funksiya grafigini chizishga bog'liq bo'lib, etarli darajada bo'lmasligi mumkin. Bu usulning mohiyati shundaki, undan foydalanayotganda qurilgan boshlang'ich grafikdan aniqligi yuqori bo'lgan ikkita ixtiyoriy $M_1(x_1^*, y_1^*)$, $M_2(x_2^*, y_2^*)$ nuqtalar olamiz va

$$\begin{cases} y_1^* = f(x_1^*, a, b) \\ y_2^* = f(x_2^*, a, b) \end{cases}$$

algebraik tenglamalar sistemasini a va b noma'lum parametrlarga nisbatan yechib, a va b parametrlarning qiymatlari aniqlanadi.

Kichik kvadratlar usuli (KKU) tanlangan nuqtalar usuliga nisbatan ancha aniq natijalar beradi, lekin bu usulda hisoblashlar ko'p bo'ladi. KKU ni keltirish uchun avval Δ_i xatolik tushunchasini kiritamiz. Δ_i xatolik y

miqdorning tajribaviy qiymati y_i bilan $f(x,a,b)$ funksiyaning x_i nuqtadagi qiymati ayirmasi kabi aniqlanadi:

$$\Delta_i = y_i - f(x_i, a, b)$$

KKU usuliga asosan a, b parametrlarning qiymatlari sifatida

$$F(a,b) = \sum_{i=1}^n (\Delta_i)^2 \rightarrow \min$$

Yani $F(a,b)$ funksiyani minimumga erishtiruvchilari olinadi. Bu funksiyani (a,b) bo'yicha minimumini topish uchun kritik nuqtalarni aniqlaymiz, yani $F(a,b)$ funksiyani a va b bo'yicha birinchi tartibli xususiy hosilalarini nolga tenglab olamiz:

$$\begin{cases} \frac{\delta F(a,b)}{\delta a} = 0 \\ \frac{\delta F(a,b)}{\delta b} = 0 \end{cases}$$

yoki

$$\begin{cases} \sum_{i=1}^n \Delta_i f'_a(x_i, a, b) = 0 \\ \sum_{i=1}^n \Delta_i f'_b(x_i, a, b) = 0 \end{cases}$$

Bu tenglamalar sistemasini a va b ga nisbatan yechib, kerakli qiymatlarni topamiz.

Agar empirik boq'liqlik uch parametrlilik $y = ax^2 + bx + c$ ko'rinishda bo'lsa,

$$F(a,b,c) = \sum_{i=1}^n (y_i - ax_i^2 - bx_i - c)^2$$

funksiyani minimumini (a,b,c) bo'yicha topish talab qilinadi. Yechilishi kerak bo'lgan tenglamalar sistemasi quyidagicha bo'ladi:

$$\begin{cases} a * \sum_{i=1}^n x_i^4 + b * \sum_{i=1}^n x_i^3 + c * \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i^2 y_i; \\ a * \sum_{i=1}^n x_i^3 + b * \sum_{i=1}^n x_i^2 + c * \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i; \\ a * \sum_{i=1}^n x_i^2 + b * \sum_{i=1}^n x_i + c * n = \sum_{i=1}^n y_i; \end{cases}$$

Bu tenglamalar sistemasini echib va $F(a,b,c)$ funksiyani shu nuqtada ekstremumga tekshirib, a,b,c -parametrlarining kerakli qiymatlarini aniqlaymiz. Shu bilan identifikatsiya masalasi to'liq echilgan hisoblanadi.

17.4. Ma'lumotlarni statistik qayta ishlash uchun Matlabning asosiy funksiyalari

Berilgan ma'lumotlar ustida statistik operatsiyalar bajarish uchun Matlabning quyidagi funksiyalarini qo'llash mumkin:

- **mean(x)** - x vektor elementlarini o'rta qiymatini qaytaradi, yoki x matritsa bo'lsa, ustunning o'rta qiymatlaridan tuzilgan qator-vektorni qaytaradi;

- **median(x)** - xuddi **mean(x)** kabi, faqat x vektorning (matritsaning) medianasini qaytaradi;

- **std(x)** - x vektor o'rta kvadratik xatoligini qaytaradi, x matritsa uchun qatorlarni o'rta kvadratik xatoliklaridan tuzilgan vektor-qatorni qaytaradi;

- **hist(x)** - x vektor elementlarini gistogrammasini chizadi. O'nta nuqta maksimum va minimum orqali masshtablanadi;

- **hist(x,n)** - n ta nuqtaning gistogrammasini maksimum va minimumga nisbatan olingan masshtabda chizadi.

Berilgan sonlarni (ma'lumotlarni) tartiblash va ajratib berish uchun quyidagi komandalar bor:

- **max(x)** - x vektor elementlarini maksimumini yoki x matritsa bo'lsa, ustunlarning maksimumlaridan iborat vektor-qatorni qaytaradi;

- **min(x)** - xuddi **max(x)** kabi, faqat minimumni qaytaradi;

- **sort(x)** - x vektor koordinatalarini o'sish tartibida joylashtiradi.

Massiv elementlarini yig'indi va ko'paytmasini hosil qilish komandalari:

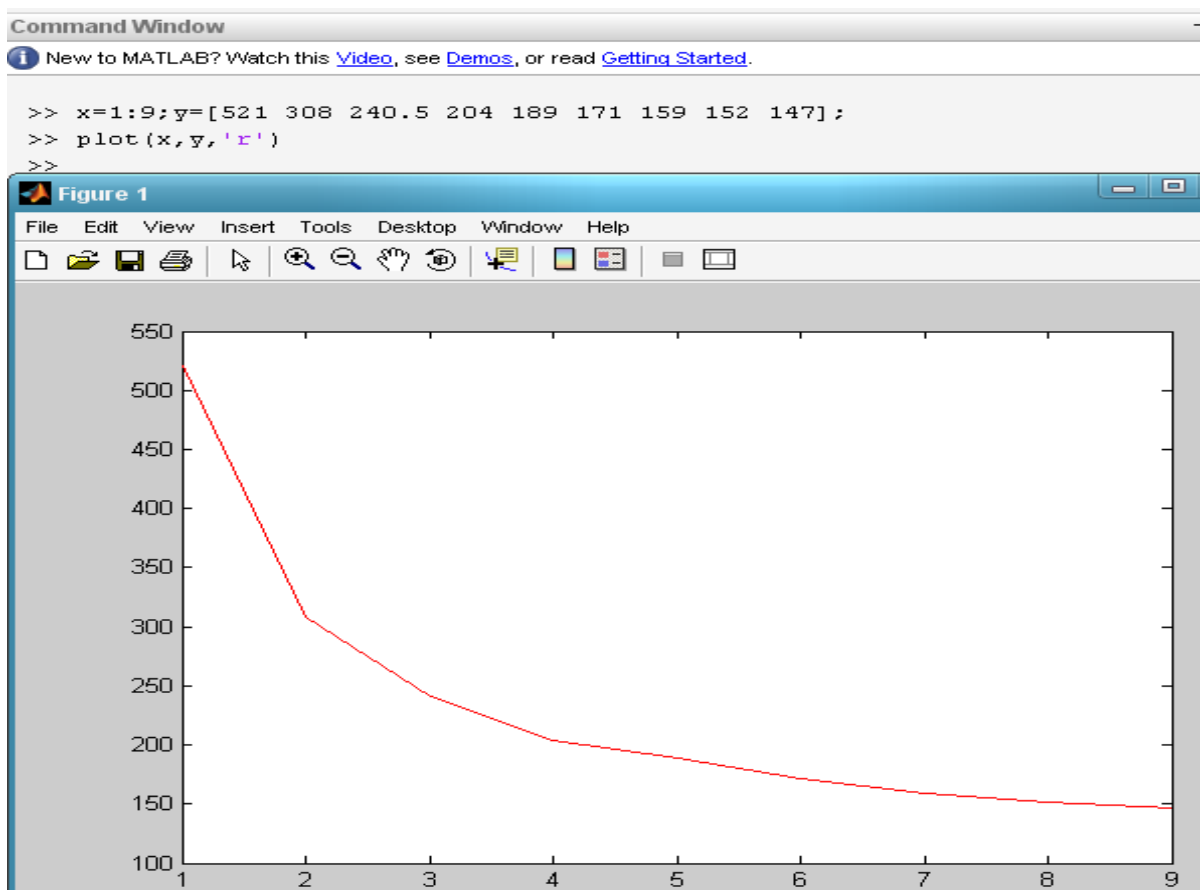
- **sum(x)** - x vektor elementlari yig'indisini qaytaradi. x matritsa bo'lsa, matritsaning mos ustun elementlari yig'indisini qaytaradi;

- **prod(x)** - xuddi **sum(x)** kabi, faqat ko'paytma qaytaradi.

Misollar. 1) Berilgan tajribaviy qiymatlar yordamida empirik bog'liqlikni aniqlang.

x	1	2	3	4	5	6	7	8	9
y	521	308	240,5	204	183	171	159	152	147

Yechish: Masalani echish uchun avval x va u o'zgaruvchilarning berilgan tajribaviy qiymatlari bo'yicha grafigini chizamiz (17.1- rasm):



17.1 - rasm. Boshlang'ich qiymatlar grafigi.

Endi x o'zgaruvchi uchun quyidagi hisoblashlarni bajaramiz:

$$x_{ar} = 5, \quad x_{geom} = 3, \quad x_{garm} = 1.8$$

Chizilgan grafikdan x ning shu qiymatlariga mos y ning qiymatlarini topamiz:

$$y_1 \approx 180, \quad y_2 \approx 242, \quad y_3 \approx 350,$$

hamda y o'zgaruvchi uchun ham huddi x niki kabi

$$y_{ar} = 334, \quad y_{geom} = 276.7, \quad y_{garm} = 229.3$$

qiymatlarni hisoblab olamiz. Endi yuqorida ko'rsatilgandek qilib, yettita ayirmaning qiymatlarini hisoblaymiz va ularning ichidan eng kichigini topamiz. U holda $\varepsilon = \varepsilon_6$ bo'ladi, demak, empirik bog'liqlik 6 – ko'rinishdagi $y = a + b/x$ giperbolik funksiya kabi olinishi mumkin. Yuqoridagi hisoblashlarni bajaruvchi Matlab dasturi quyidagicha bo'ladi:

```
x=[1:9]; y=[ 521 308 240.5 204 189 171 159 152 147];
n=length(x);
hold on
plot(x,y)
grid
x_ar=(x(1)+x(n)/2); x_geom=sqrt(x(1)*x(n));
```

```

x_garm=(2*x(1)*x(n)/( x(1)+x(n)));
disp(['x_ar=',num2str(x_ar)]);
disp(['x_geom=',num2str(x_geom)]);
disp(['x_garm=',num2str(x_garm)]);
pause
plot

```

Endi empirik boq'liqlik parametrlari a va b koeffitsiyentlarni tanlangan nuqtalar usuli bilan aniqlaymiz. Buning uchun ikkita $(1;521)$ va $(4;204)$ nuqtani tanlaymiz. U holda hosil bo'lgan

$$a+b/1 = 521,$$

$$a+b/4 = 204$$

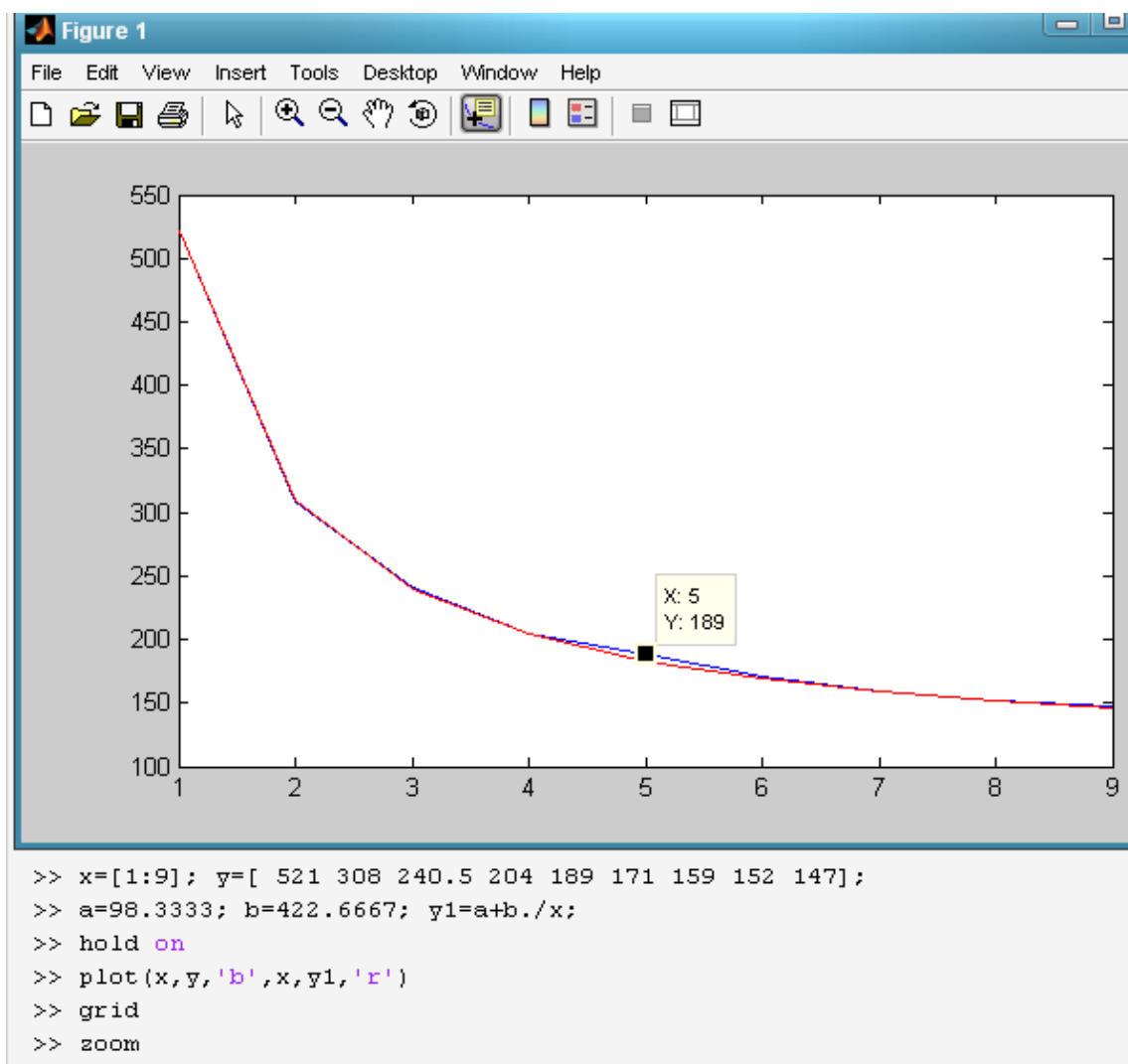
tenglamalar sistemasini echib, a , b larning taqribiy qiymatlarini topamiz:

$a \approx 98.3333$, $b \approx 422.6667$. Hosil bo'lgan $y = 98.3333+422.6667/x$ funksiya grafigini chizamiz va uni boshlanq'ich qiymatlar grafigi bilan solishtiramiz. Bunday solishtirish uchun, ikkita grafikni bir oynada hosil qiluvchi Matlabning quyidagi dasturidan foydalanamiz:

```

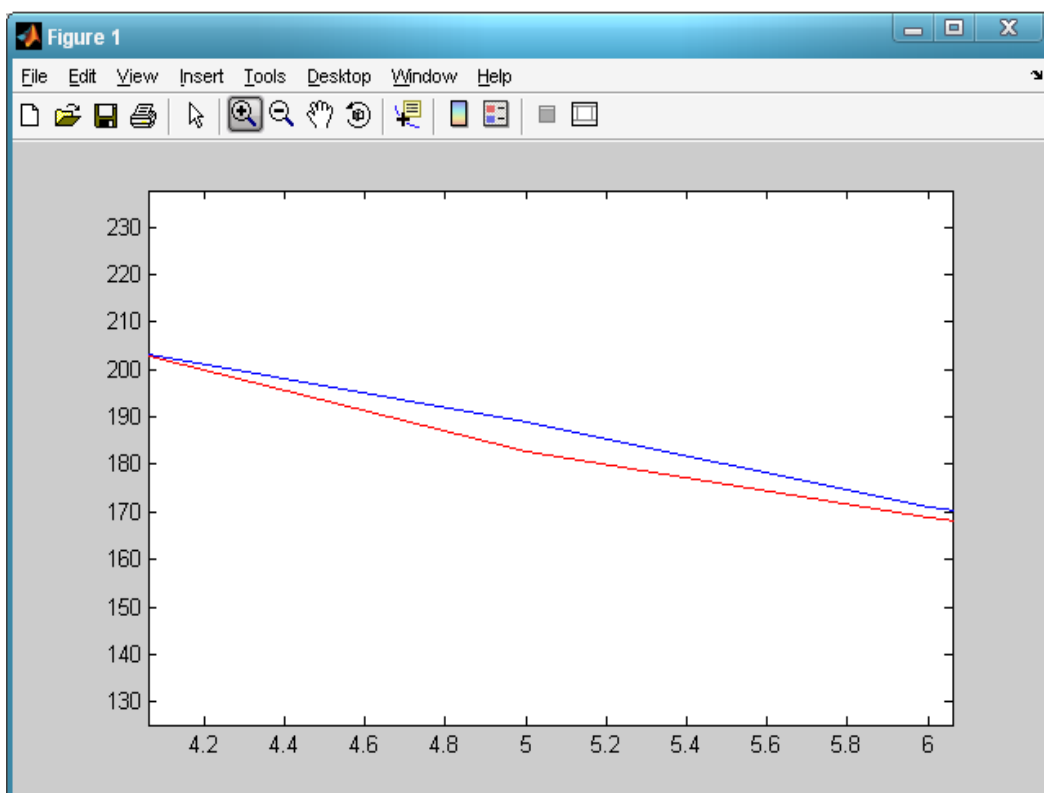
x=[1:9];
y=[ 521 308 240.5 204 189 171 159 152 147];
a=98.3333; b=422.6667;
y1=a+b/x;
hold on
plot(x,y,'b',x,y1,'r')
zoom

```



17.2-rasm. Solishtirish grafigi.

Bu grafikni masshtablash yordamida berilganlar va aniqlangan funksiya grafiklari orasidagi farqni hamda xatolikni ko'rish mumkin(17.3-rasm).



17.3-rasm. Masshtablab solishtirish.

2) Empirik boq'liqlik $y=ax^2+bx+c$ bo'lganda KGU yordamida a, b, c parametrlarni aniqlang. Boshlanq'ich qiymatlar quyidagi jadvalda berilgan:

x	2	2,2	2,4	2,6	2,8	3
y	0,3	0,3	0,3	0,4	0,4	0,4
	010	424	802	150	472	771

Qo'yilgan masalani yechish uchun quyidagi MATLAB dasturidan foydalanamiz:

```
x=2:.2:3; y=[.3010 .3424 .3802 .4150 .4472 .4771];
A(1,1)=sum(x.^4);
A(1,2)=sum(x.^3);
A(1,3)=sum(x.^2);
B(1,1)=sum((x.^2).*y);
A(2,1)= A(1,2);
A(1,3)=A(1,3);
A(2,3)=sum(x);
B(2,1)=sum(x.*y);
A(3,1)= A(1,3);
A(3,2)= A(2,3);
A(3,3)=length(x);
B(3,1)=sum(y);
```

`% A*x=B u holda x=A\B bo`ladi`

`x=A\B;`

`disp(['a=',num2str(x(1))]);`

`disp(['b=',num2str(x(2))]);`

`disp(['a=',num2str(x(3))]);`

U holda a , b , c koeffitsiyentlar uchun sonli qiymatlar hosil qilamiz:

$$a=-0.03567, \quad b=0.35402, \quad c=-0.26414.$$

Demak, izlanayotgan empirik funksiya quyidagicha bo`ladi:

$$y=-0.03567*x^2+0.35402*x-0.26414.$$

Nazorat savollari

1. Boshlang'ich ma'lumotlarni birlamchi qayta ishlash masalasining qo'yilishi qanday?
2. Empirik bog'liqlikning strukturali identifikatsiya masalasi algoritmi qanday?
3. Tanlangan nuqtalar usulini keltiring.
4. KKU ni tushuntirib bering.
5. Matlabda ma'lumotlarga statistik qayta ishlash funksiyalarini barchasini keltiring.

18. BIR VA KO'P O'ZGARUVCHILI FUNKSIYALAR UCHUN OPTIMALLASHTIRISH

18.1. Funksiyalar uchun optimallashtirish masalasining qo'yilishi

Juda ko'p nazariy va amaliy masalalarni hal qilishda bir nechta o'zgaruvchiga boq'liq bo'lgan funksiyalarning ekstremumini (maksimum yoki minimum) topish masalasiga duch kelinadi (masalan, parametrik identifikatsiya masalasida). Bunday funktsiyani umumiy holda $f(x_1, x_2, \dots, x_n)$ ko'rinishida yozib, $x = (x_1, x_2, \dots, x_n)$ vektorni kiritsak, u holda $f(x)$ funksiya uchun ekstremumni (ma'lum bir A to'plamda) topish quyidagicha qo'yiladi:

x vektorning berilgan (aniqlangan) A to'plamga tegishli shunday x^* qiymatini topingki, u uchun

$$\max_{x \in A} f(x) = f(x^*)$$

tenglik o'rinli bo'lsin. Albatta, bu nuqtada $f(x)$, $x \in A$, funksiya uchun $f(x) \leq f(x^*)$, $x \in A$, tengsizlik o'rinli bo'ladi. x^* nuqta funktsiyaning maksimum nuqtasi, $f(x^*)$ esa funktsiyaning maksimum qiymati deyiladi. Huddi shunga o'xshash minimum nuqta haqida ham gapirish mumkin. Umuman olganda, maksimum va minimum masalalarini birinchisini ikkinchisiga keltirish mumkin. Masalan, $f(x)$, $x \in A$, funktsiyani maksimumini topish masalasi $g(x) = -f(x)$, $x \in A$, funktsiyaning minimumini topishga ekvivalentdir.

Funktsiyaning minimumini yoki maksimumini topish optimallashtirish masalasi deb ataladi.

18.2. Funksiyalar uchun optimallashtirish masalasini yechish usullari

Matematikada har xil tipdagi funktsiyalarni optimallashtirish usullari juda ham ko'p. Ularni masalani yechishga talqin qilish bo'yicha ikkita guruhga ajratish mumkin.

Birinchi guruhga masalani hal qilish uchun qo'llaniladigan bilvosita usullarni kiritish mumkin. Bu holda optimallashtirish masalasi ko'p o'zgaruvchili funktsiyalar uchun x^* nuqtada ekstremum shartining natijasi bo'lgan chiziqli yoki chiziqsiz tenglamalar sistemasini yechimini topishga keltiriladi. Bizga ma'lumki, ekstremum nuqtada funktsiyaning barcha birinchi tartibli xususiy hosilalari nolga teng bo'ladi:

$$\frac{\partial f}{\partial x_i} \Big|_{x=x^*} = 0, \quad i=1,2,\dots,n.$$

Shu tenglamalar sistemasini yechib, ekstremum bo'lishi mumkin bo'lgan nuqta aniqlanadi. Bundan tashqari birinchi guruh usullariga vatarlar, Nyuton usullarini va boshqalarni kiritish mumkin.

Bu usullarning asosiy kamchiliklariga chiziqsiz tenglamalar sistemasini yechishdagi murakkabliklar kiradi. Shuning uchun, ko'pincha optimallashtirish masalasini amalda yechish uchun taqribiy usullar qo'llaniladi. Bu holda optimallashtirish masalasini yechish uchun shunday

$$x^0, x^1, \dots, x^n, \dots \text{ vektorlar ketma-ketligi tuziladiki, ular uchun} \\ f(x^0) < f(x^1) < \dots < f(x^n) < \dots \quad (f(x^0) > f(x^1) > \dots > f(x^n) > \dots)$$

tengsizlik o'rinli bo'lsin. Natijada, ma'lum qadamdan keyin ekstremum nuqtaning taqribiy qiymati topiladi. Umuman olganda, boshlanq'ich nuqta x_0 ixtiyoriy bo'lishi mumkin, lekin uni tanlashda funksiya va uni ekstremumi haqida barcha ma'lumotlarni ishlatib, x_0 ni ekstremum nuqtaga iloji boricha yaqin qilib tanlash maqsadga muvofiqdir.

18.3. Optimallashtirish masalasini echish uchun MATLAB funksiyalari

Optimallashtirish masalasini yechish uchun Matlab paketi yadrosidagi va maxsus Optimization kutubxonasiidagi (vositalar to'plami) funksiyalardan foydalanish mumkin. Bu funksiyalarni ko'rishdan avval Matlabda ishtirok etuvchi qo'shimcha element –foptions1 massivi bilan tanishaylik. Bu massivda <standart parametrlar> (<параметры по умолчанию>) deb nomlanuvchi va optimizatsiya proseduralarida foydalaniladigan parametrlar saqlanadi. Ushbu massivning bir elementini ko'rib chiqamiz:

- options(1)-akslantirish parametri (avtomatik tarzda 0 ga teng), 1 qo'yilganda ba'zi natijalarni akslantiradi;
- options(2)-x uchun hisoblashlar to'xtatilishining aniqligi; avtomatik tarzda 1e-4;
- options(3)-F uchun hisoblashlar to'xtalishining aniqligi; avtomatik tarzda 1e-4;
- options(4)-chegara buzilishida uzish kriteriysi; avtomatik tarzda 1e-6;
- options(5)-algoritm: strategiya: har doim ham ishlatilavermaydi;
- options(6)-algoritm: Optimizator: Har doim ham ishlatilavermaydi;

- options(7)-algoritm: Chiziqli qidiruv algoritmi; avtomatik tarzda 0;
- options(8)-Liyambda funksiyaning qiymati;
- options(9)-agar foydalanuvchi taklif qilgan gradientlarni tekshirish kerak bo'lsa, bu parametrga 0 qo'yiladi;
- options(10)-funksiya va chegaralarni baholashlar soni;
- options(11)-funksiya gradientini baholashlar soni;
- options(12)-chegaralarni baholashlar soni;
- options(13)-tenglikka qo'yilgan chegaralar soni;
- options(14)-funksiyaning maksimal baholashlar soni;
- options(15)-maqsadli funksiyaning maxsus maqsadlar uchun ishlatish;
- options(16)-chekli ayirmali gradientlar uchun o'zgaruvchilarning minimal o'zgarishi;
- options(17)-chekli ayirmali gradientlar uchun o'zgaruvchilarning maksimal o'zgarishi;
- options(18)-qadam uzunligi (avtomatik tarzda ≤ 1);

Har xil optimizatsiya jarayonlari uchun bu parametrlardan har xillari ishlatiladi. Shuning uchun konkret optimizatsiya jarayoni uchun qanday parametr berilgan bo'lishi va qanday parametr ma'lum natijani qaytarishini alohida aytib o'tish kerak bo'ladi.

Parametrlar avvaldan aniqlab olingandan so'ng, funksiyaning optimallashtirish jarayoniga o'tsa bo'ladi. MATLAB yadrosida optimallashtirish masalasini yechish uchun bir nechta funksiyalar mavjud bo'lib, ular quyidagilardir: bir o'zgaruvchili funksiyalar uchun **fminbnd** funksiyasi; ko'p o'zgaruvchili funksiyalar uchun esa **fminsearch** funksiyasidir.

fminbnd funksiyasi quyidagi formatlarga ega:

- **fminbnd(ffun,x1,x2)** – $x_1 < x < x_2$ intervalda **ffun(x)** funksiyaga lokal minimumni beruvchi x ning qiymatini qaytaradi.

- **fminbnd(ffun,x1,x2,options)** - yuqorida keltirilgan funksiya bilan o'xshash, lekin **options** vektoridan **tolX**, **maxfuneval**, **maxiter**, **display** parametrlarini qo'llaydi, bu parametrlar oldindan **optimset** komandasi orqali o'rnatilgan bo'ladi (batafsil ma'lumot uchun **lsqnonneg** komandasiga qarang)

- **fminbnd(ffun,x1,x2,options,P1,P2,...)** – yuqoridagi tavsif bilan o'xshash, lekin maqsad funksiyaga qo'shimcha **P1,P2,...** argumentlarni uzatadi: agar hisoblash parametrlarini avtomatik o'rnatilgan holdagi

ko'rinishida qo'llash kerak bo'lsa, u holda P1,P2 oldidan bo'sh massiv “[]” kiritish kerak bo'ladi (options o'rniga).

- **[x,fval] = fminbnd(...)** – fval maqsad funksiyani minimum nuqtadagi qiymatini qo'shimcha ravishda qaytaradi.

- **[x,fval,exitflag] = fminbnd(...)** – agar funksiya options.tolX ni qo'llash bilan mos kelsa, exitflag parametrini 1 qiymat bilan qaytaradi; agar options.maxiter iteratsiyalarning maksimal soniga erishilgan bo'lsa, exitflag parametrini 0 qiymat bilan qaytaradi.

Keltirib o'tilgan tavsiflarda quyidagi belgilar qo'llanilgan: [x1,x2] – interval, unda funksiya minimumi qidiriladi; P1,P2 ... – qo'shimchalar, x-funksiya argumenti; ffun – satr, o'zida funksiyaning nomini saqlaydi, funksiya esa o'z navbatida minimallashtiriladi; options – hisoblash parametrlarining vektori.

fminbnd funksiyani berilish formasiga boq'liq ravishda minimumni hisoblash ma'lum “tilla kesim” yoki “parabolik interpolyatsiya” metodlari orqali amalga oshiriladi.

Misol:

```
>> options=optimset('tolX',1.e-10);...  
[x]=fminbnd(@cos,3,4,options)  
x = 3.1416
```

fminsearch funksiyasi quyidagi formatlarga ega:

- **fminsearch(fun, xo)** – fun(x) funksiyaning xo yaqinida lokal minimum beruvchi x vektorni qaytaradi, xo skalyar ham, vektor ham (bir o'zgaruvchili funksiyaning minimallashtirish kesmasi) yoki matritsa (bir necha o'zgaruvchili funksiya uchun) bo'lishi mumkin;

- **fminsearch(fun,x0,options)** – yuqorida keltirilgan funksiya bilan o'xshash, lekin options parametrlar vektorini fminbnd funksiya kabi qo'llaydi;

- **fminsearch(fun,x0,options,P1,P2,...)** – yuqorida berilgan funksiya o'xshash, lekin minimallashtirilayotgan fun(x,P1,P2,...) funksiya qo'shimcha P1,P2,... argumentlarni beradi. Agar hisoblash parametrlarini avtomatik o'rnatilgan holda qo'llash kerak bo'lsa, u holda P1,P2 oldida options o'rniga “[]” belgisini kiritish kerak bo'ladi;

- **[x,fval] = fminsearch(...)** – qo'shimcha ravishda fval maqsad funksiyasining minimum nuqtadagi qiymatini qaytaradi;

- **[x,fval,exitflag] = fminsearch(...)** – qo'shimcha holda exitflag parametrini qaytaradi; agar iteratsiya jarayoni options.tolX bilan mos

tushsa, musbat; iteratsiya olingan yechim X ga yaqinlashmasa, manfiy; options.maxiter iteratsiya maksimal sonidan oshgan bo'lsa, 0 bo'ladi.

- **[x,fval,exitflag,output] = fminsearch(...)** – output strukturasi (yozuv)ni qaytaradi;

- **output.algorithm** – ishlatilgan algoritm;
- **output.funcSount** – maqsad funksiyani baholashlar soni;
- **output.iterations** – amalga oshirilgan iteratsiyalar soni;

Ko'p o'zgaruvchili funksiyalarni optimallashtirish uchun keltirilgan Matlab funksiyalari simpleks-metodning bir turi bo'lgan Nelder-Mid metodi asosida qurilgan. Bu metod ko'p o'zgaruvchili funksiyani minimallashtirishda eng yaxshi to'g'ridan-to'g'ri usul hisoblanadi, bunda gradientni hamda funksiyani hisoblash talab qilinmaydi. n o'lchamli fazoda berilgan $n+1$ cho'qqi asosida simpleksni qurishga keltiriladi. Ikki o'lchamli fazoda simpleks uchburchak shaklida bo'ladi, uch o'lchamli fazoda esa – piramida ko'rinishida bo'ladi. Iteratsiyaning har bir qadamida echimning yangi nuqtasi simpleksning ichida yoki simpleksga yaqin joydan tanlanadi. U simpleksning biron-bir cho'qqisi bilan solishtiriladi. Bu nuqtaga yaqin simpleks cho'qqisi shu nuqta bilan o'rin almashadi. Shu tariqa simpleks qayta joylashadi va odatda yangiroq, aniqroq echim nuqtasini topishga imkon beradi. Yechish jarayoni simpleks o'lchamlari barcha o'zgaruvchilari bo'yicha berilgan echim xatoliklaridan kam bo'lmagunga qadar qaytarilaveradi.

Matlab ning eski versiyalarida ishlash tajribasiga ega foydalanuvchilar, optimallashtirish funksiyasidagi nomlar orasidagi farqni inobatga olishlari kerak. Ular quyida keltirib o'tilgan:

4 - versiyagacha va undan past	5 - versiya va undan yuqori
fmin	fminbnd
fmins	fminsearch
foption	soptimget, optimsetfzero
nls	lsqnonneg,fminunc

18.4. Funksiya ekstremumini topishga doir misollar

1. $y=\exp(-x)*\sin(3*\pi*x)$ funksiyaning $[0,2]$ oraliqdagi minimumini toping.

M-fayl tuzib olamiz :

```
function y=shux(x)
```

```
y=exp(-x)*sin(3*pi*x);
```

Endi komandalar oynasidan murojaat qilamiz:

```
>> [x,y]=fminbnd('shux',0,2)
```

```
x =
```

```
1.1555
```

```
y =
```

```
-0.3132
```

2. $y = x \cdot \sin(x)$ funksiyaning $[-10,10]$ oraliqdagi minimumini toping.

```
>> [x,y]=fminbnd('x*sin(x)',-10,10)
```

```
x =
```

```
-4.9132
```

```
y =
```

```
-4.8145
```

3. $u = \sin(x) + \cos(x)$ funksiyaning $[-2,10]$ oraliqdagi minimumini toping.

Inline funksiyadan foydalanib topamiz.

```
>> func=inline('sin(x)+cos(x)')
```

```
func =
```

```
Inline function:
```

```
func(x) = sin(x)+cos(x)
```

```
>> fminbnd(func,-2,10)
```

```
ans = 3.9270
```

```
>> func(3.9270)
```

```
ans = -1.4142
```

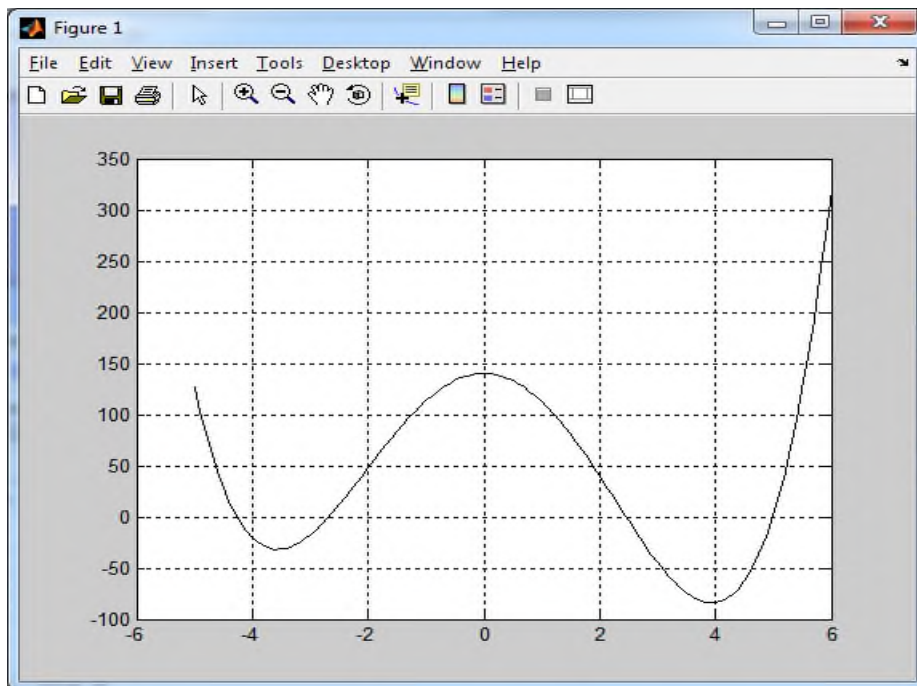
4. $y = x^4 - 0.5x^3 - 28x^2 + 140$ funksiyaning $[-5;6]$ oraliqdagi minimumini topilsin.

Natija aniq va ko'rgazmali namoyish etilishi uchun avval quyidagi buyruqlardan foydalanib, funksiyaning grafigini chizib olamiz:

```
>> x=-5:0.1:6;
```

```
>> y=x.^4-0.5*x.^3-28*x.^2+140;
```

```
>> plot(x,y,'-k'), grid
```



18.1-rasm. $y=x^4-0.5x^3-28x^2+140$ funksiya grafigi

Endi m-fayl - funksiyanı yazib olamiz:

```
function y=fun_min(x)
```

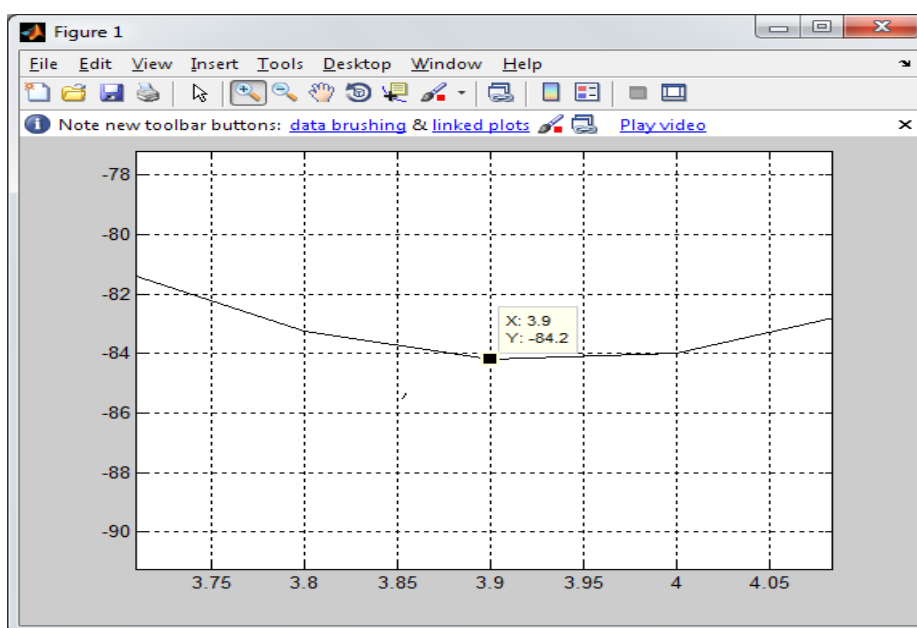
```
y=x.^4-0.5*x.^3-28*x.^2+140;
```

So'ng buyruqlar oynasida grafikdan foydalangan holda kerakli oraliqlarni ko'rsatib, quyidagi buyruqlarni kiritamiz:

```
>>[x,y]=fminbnd(@fun_min,2,6)
```

```
x = 3.9339
```

```
y = -84.2624
```



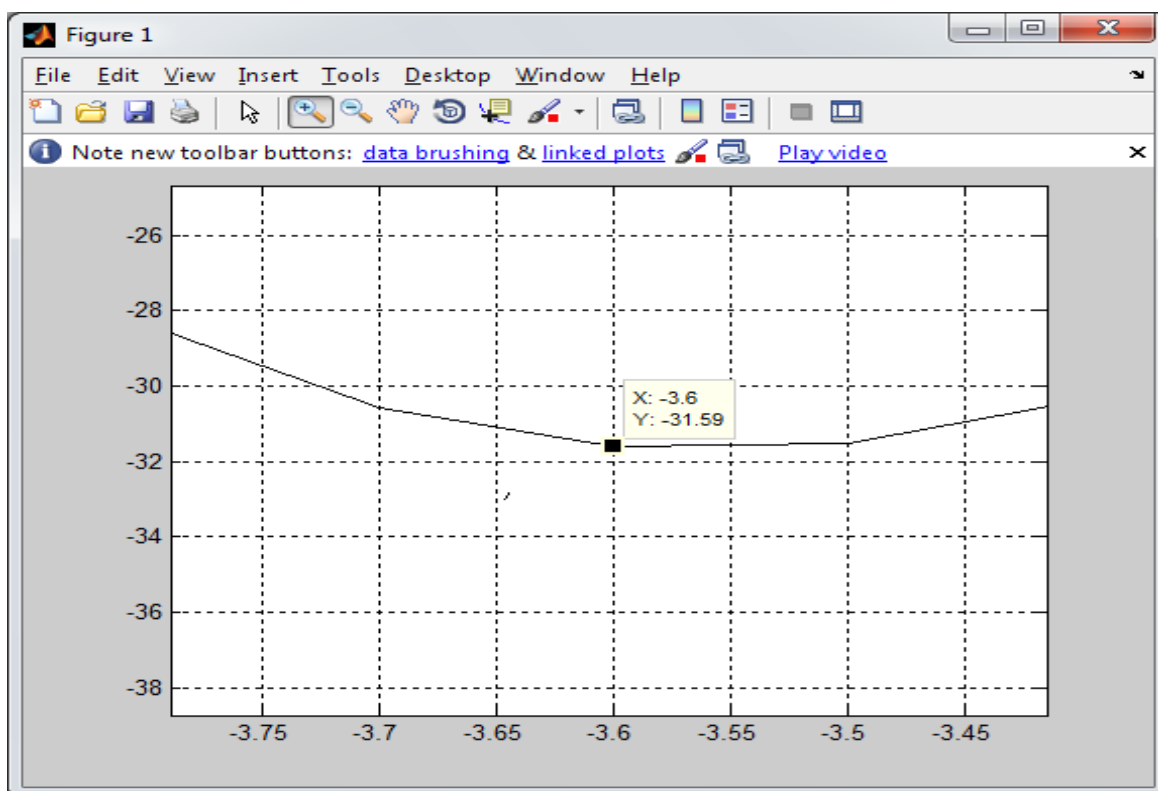
18.2 -rasm. $[2,6]$ oraliqdagi minimum

Endi [-5;2] oraliqda minimum qidiramiz:

```
>>[x,y]=fminbnd(@fun_min,-5,-2)
```

```
x = -3.5589
```

```
y = -31.6817
```



18.3-rasm. [-5,-2] oraliqda funksiya minimumi.

Grafikdan ko'rinib turibdiki, [-5;6] oraliqda qaralayotgan funksiya maksimum qiymatga ham ega. Bu qiymatni topish uchun funksiya oldiga “-” ishora qo'yib, keyin fminbnd funksiyasidan foydalanamiz:

```
>>[x,y]=fminbnd('-(x.^4-0.5*x.^3-28*x.^2+140)',-5,6)
```

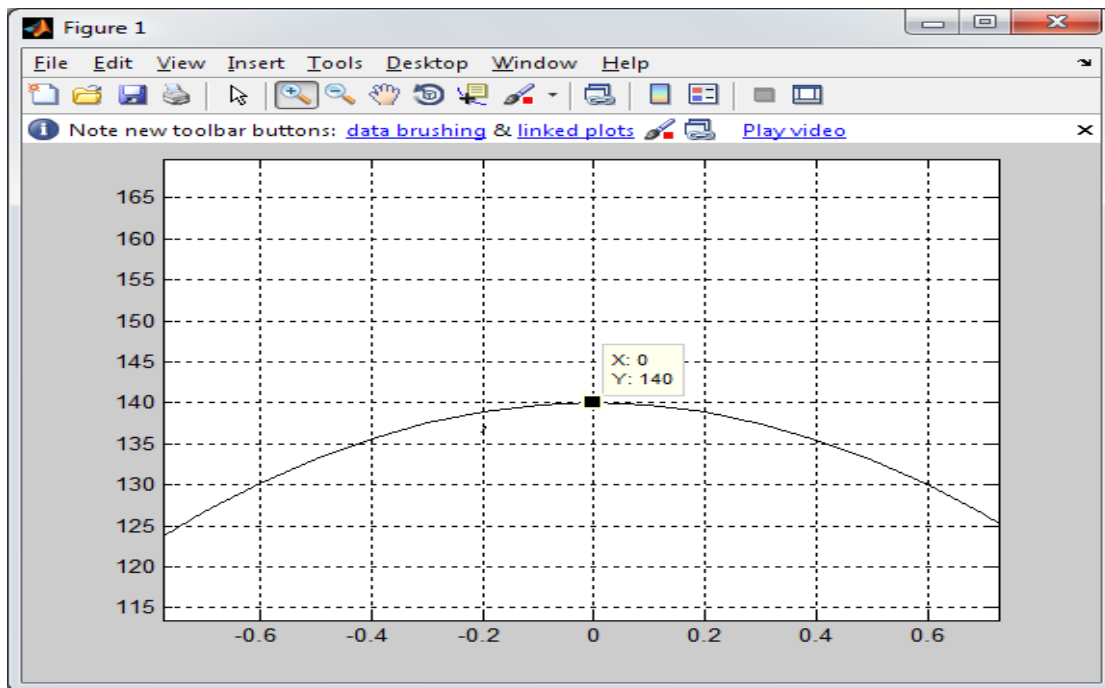
```
x =
```

```
-1.4521e-005
```

```
y =
```

```
-140.0000
```

Demak, qaralayotgan funksiya hamizning maksimumi $x=-1.4521e-005$ da erishiladi va $y=140$ qiymat bo'ladi (chunki fminbnd funksiyadan foydalanayotganda “-” ishora qo'yilgan edi).



18.4-rasm. $[-5,6]$ oraliqdagi maksimum.

Ta'kidlash joizki, agar f_{minbnd} funksiyani birdaniga $[-5;6]$ oraliqda qo'llasak, faqat bitta $x=3.9339$ nuqtadagi $y=-84.2624$ minimum qiymatni beradi (shuning uchun mashq sifatida yuqorida qaralgan 1-3 misollarni tekshiring!).

5. $f=\sin(\pi*x)*\sin(\pi*u)$ funksiya minimumini $[1.4,2.6] \times [1.4,2.6]$ to'plamda toping. M-fayl tuzib olamiz:

```
function f=dilf(v)
```

```
x1=v(1); x2=v(2); f=sin(pi*x1).*sin(pi*x2);
```

Endi komandalar oynasidan murojaat qilamiz:

```
>> [x,f]=fminsearch('dilf',[1.4 2.6])
```

```
x = 1.5000 2.5000
```

```
f = -1.0000
```

6. $f(x,y)=\sqrt{x^2+y^2}$ funksiyaning minimumi topilsin.

m-fayl funksiya yaratamiz:

```
function f=funs_min(x)
```

```
f=sqrt(x(1).*x(1)+x(2).*x(2));
```

Buyruqlar oynasidan murojat qilamiz:

```
>>[x f]=fminsearch('funs_min',[-2 2])
```

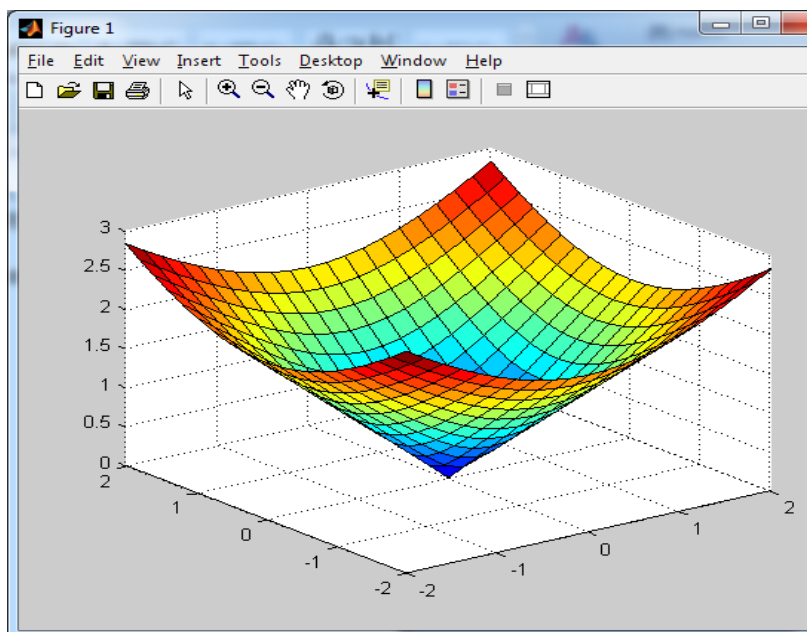
```
>>[x y]=meshgrid(-2:0.2:2, -2:0.2:2);
```

```
>>z=sqrt(x.^2+y.^2); surf(x,y,z);
```

```
x = 1.0e-004 *
```

```
0.4133 -0.1015
```


$$f = 4.2559e-005$$



18.5-rasm. $f(x,y)=\sqrt{x^2 + y^2}$ funksiya grafigi.

Xulosa qilib shuni aytish mumkinki, optimallashtirish masalalarini yechishda Matlab dasturining imkoniyatlari juda katta (masalan, lsqnonlin, fminmax, fminunc, fmincon funksiyalari ham mavjud)

18.5. Rozenbrok test funksiyasini minimallashtirish

fminsearch funksiyasini qo'llanilishiga misol sifatida klassik test funksiyasining minimumini topish masalasini ko'rsak bo'ladi. Rozenbrok funksiyasining minimum nuqtasi "yassi tub" li "jar" likda joylashgan:

$$rb(x1, x2, a) = 100*(x2 - x1^2)^2 + (a - x1)^2.$$

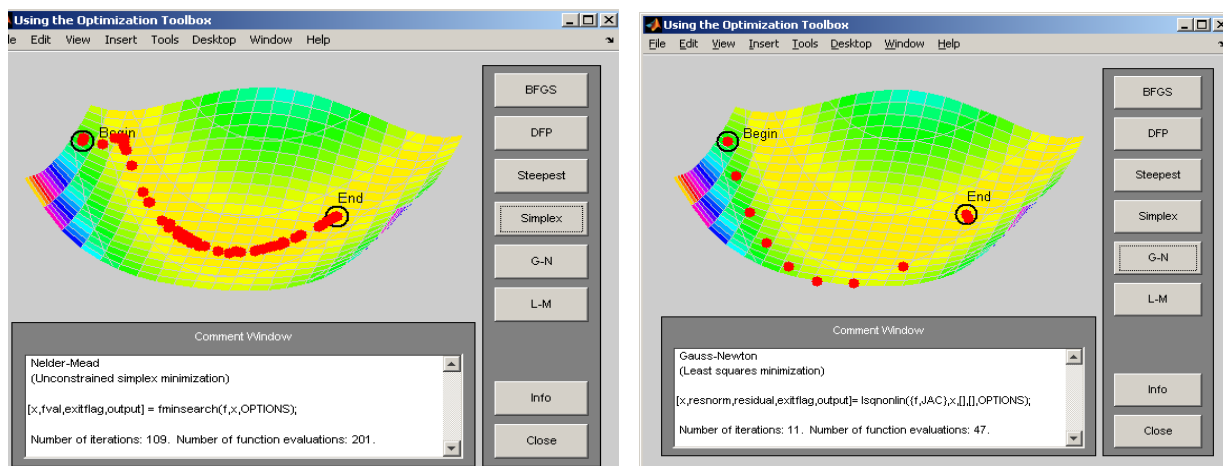
Bu funksiyaning minimal qiymati nolga teng va unga (a, a^2) nuqtada erishadi. Misol tariqasida $x1$ va $x2$ qiymatini $(-1.2, 1)$ nuqtada aniqlashtiramiz. **(rb.m)** faylida funksiyani kiritamiz:

```
% Rozenbrokning test funksiyasi
function f=rb(x,a)
if nargin<2 a=1; end
f=100*(x(2)-x(1)^2)^2+(a-x(1))^2;
>> options=optimset('tolX',1.e-6);
[xmin, opt, rosexflag,rosout]=fminsearch(@rb,[-1.2 1],options)
xmin = 1.0000 1.0000
opt = 4.1940e-014
rosexflag = 1
rosout = iterations: 101
```

funcCount: 189

algorithm: 'Nelder-Mead simplex direct search'

Ko'p o'zgaruvchili funktsiyani minimallashtirish mohiyatini yanada chuqurroq tushunish uchun Demos namoyish kutubxonasidagi misollarni ko'rib chiqish tavsiya qilinadi.



18.6 - rasm. Demos namoyish kutubxonasi illyustratsiyasi.

18.6-rasmda Rozenbrok funktsiyasini simpleks va Gauss-Nyuton metodi orqali minimallashtirishning grafik illyustratsiyasi ko'rsatilgan.

18.6. Ko'p o'zgaruvchili funktsiyani minimallashtirishning boshqa usullari

Bir necha o'zgaruvchili funktsiyani minimallashtirishda Optimization Toolbox paketidan Matlab **fminunc** va **lsqnonlin** funktsiyalarini qo'llash mumkin. **fminunc** funktsiya `optimset` komandasi orqali avvaldan kiritilgan maqsad funktsiyasining yaqinlashish chegarasini, gradientlarning `options.gradobj` vektorini, Gees matritsasini, Gees matrisatsining ko'paytirish funktsiyasini yoki maqsad funktsiyasining Gees matrisatsi siyraklik gradientini qo'llash imkonini beradi. **lsqnonlin** komandasi eng kichik kvadratlar metodi va odatda minimizatsiyada iteratsiyalarning eng kichik sonini beradi. Quyida Rozenbrok funktsiyasini minimizatsiya qilish uchun yuqorida keltirilgan komandalarni amalda ko'rsatamiz:

```
>> options=optimset('tolX',1e-6,'TolFun',1e-6);  
>> [xmin, opt, exflag, out, grad, hessian ]=fminunc(@rb,[-1.21 2],  
options)
```

```

Command Window
>> options=optimset('tolX',1e-6,'TolFun',1e-6);
>> [xmin, opt, exflag, out, grad, hessian]=fminunc(@rb,[-1.2 1], options)
Warning: Gradient must be provided for trust-region method;
        using line-search method instead.
> In fminunc at 281
Optimization terminated: relative infinity-norm of gradient less than options.TolFun.

Computing finite-difference Hessian using user-supplied objective function.

xmin =

    1.0000    1.0000

opt =

    2.8336e-011

exflag =

     1

out =

    iterations: 36
    funcCount: 138
    stepsize: 1
    firstorderopt: 1.9100e-005
    algorithm: 'medium-scale: Quasi-Newton line search'
    message: [1x85 char]

grad =

    1.0e-004 *
    -0.1910
     0.0872

hessian =

    802.2845 -400.0223
   -400.0223  200.0000

```

18.7 – rasm. Rozenbrok funksiyasini minimumi.

firstorderopt – maqsad funksiyasi gradientining birinchi norma uchun aniqlangan minimum nuqtadagi optimallik o'lchovi:

```
>>options=optimset('tolX',1e-6, 'maxFunEvals',162);
```

```
>> [xmin, opt]=lsqnonlin(@rb,[-1.2 1],[0 1e-6],[0 1e-6],options)
```

Warning: Large-scale method requires at least as many equations as variables; switching to line-search method instead. Upper and lower bounds will be ignored.

> In S:\MATLABR12\toolbox\optim\private\lsqncommon.m at line 155

In S:\MATLABR12\toolbox\optim\lsqnonlin.m at line 121

Maximum number of function evaluations exceeded

Increase OPTIONS.maxFunEvals

```
xmin = 0.6120      0.3715
opt = 0.1446
```

E'tibor bersangiz, lsqnonlin funksiyasi kutilgan natijani bermadi. Iteratsiya sonini chegaradan o'tib ketganligi haqida ma'lumot chiqdi, xmin qiymati esa haqiqatdan ancha yiroq.

Ko'p o'zgaruvchili funksiyani minimumini qidirish uchun fminsearch funksiyasidan foydalanib ko'rish mumkin. Misol uchun, m-fayl funksiya ko'rinishida three_var uch o'zgaruvchili funksiyani aniqlab olamiz:

```
function b = three_var(v)
x = v(1); y = v(2); z = v(3);
b = x.^2 + 2.5*sin(y) - z^2*x^2*y^2;
```

Endi esa o'zgaruvchilarning turli xil boshlanq'ich qiymatlarida ushbu funksiyaning minimumini topamiz:

```
>> v = [-0.6 -1.2 0.135];
a = fminsearch(@three_var,v)
a = 0.0000      -1.5708      0.1803
>> three_var(a)
ans = -2.5000
>> v = [-1 -1.2 0];
>> a = fminsearch(@three_var,v)
a = 0.0000      -1.5708      0.0015
>> three_var(a)
ans = -2.5000
>> v = [-1 -1.2 0.2];
>> a = fminsearch(@three_var,v)
a = 0.0000      -1.5708      0.25
>> three_var(a)
ans = -2.5000
```

Yuqoridagi misolga e'tibor bersangiz, dastlabki ikkita o'zgaruvchilari bo'yicha minimumi o'zgaruvchilarning har xil boshlanq'ich qiymatlarida bir xil. Ammo uchinchi o'zgaruvchi bo'yicha minimum o'zgaruvchilarning boshlanq'ich qiymatlariiga boq'liq. Shunga qaramasdan funksiyaning o'z qiymati barcha hollarda bir xil.

Endi quyida biron-bir echimga erishilmaydigan misol ko'ramiz:

```
>> v = [-1 -1.2 1];
>> a = fminsearch(@three_var,v)
```

Exiting: Maximum number of function evaluations has been exceeded – increase MaxFunEvals option.

Current function value: -Inf

a = 1.0e+051 *

-0.5630 -7.3469 3.8861

Demos misollar bibliotekasida siz ushbu funksiyalar qo'llanishiga doir qo'shimcha misollar topishingiz mumkin. Optimization Toolbox va Genetic Algorithm and Direct Search Toolbox kengaytma paketlarida ko'p o'zgaruvchili funksiya ekstremumini hisoblash imkoniyatini beruvchi yangi kuchli vositalar mavjud, uning ichida esa o'z navbatida optimallashtirish masalalarining yangi kuchli yechim algoritmlari, jumladan genetik va to'qridan-to'qri izlash algoritmlari bor. Ular, xususan, murakkab funksiyalarning global ekstremumlarini topish va boshqalar uchun qo'llanilishi mumkin.

18.7. Optimizatsion kutubxonaning imkoniyatlari

Optimizatsion kutubxonasi chiziqli va chiziqli bo'lmagan funksiyalarni optimallashtirishga mo'ljallangan bo'lib, bu kutubxona quyidagi xossalarga ega:

- chiziqli bo'lmagan funksiyalarni shartsiz optimallashtirish;
- kichik kvadratlar usuli va chiziqli bo'lmagan interpolatsiya;
- chiziqli bo'lmagan tenglamalar echimi;
- chiziqli dasturlash;
- kvadratik dasturlash;
- chiziqli bo'lmagan funksiyalarni shartli minimizatsiya qilish;
- minimaks usuli;
- ko'p kriteriyali optimallashtirish;

Bu kutubxonada quyidagi algoritmlar ishlatiladi:

- shartsiz optimallashtirish: Nelder-Mid simpleks qidiruv usuli;
- shartli ko'p kriteriyali optimallashtirish va minimaks usuli: ketma-ket kvadratik dasturlash usulining har xil variantlari;
- chiziqli va kvadratik dasturlash usullari: proeksiyalar usuli;
- optimallashtirish usuli va chiziqli qidiruv strategiyasini tanlash imkoniyati borligi.

Undan tashqari, kutubxonada bitta masalani bir nechta usullar yordamida yechish mumkinligini ko'rsatuvchi misollar ham mavjuddir.

Mustaqil ishlash uchun misollar

Funksiyalarning ekstremumlarini toping

- 1) $y = x^3 - 3x^2$, [-1 ; 4]
- 2) $y = x \ln x$, [0,1 ; 1]
- 3) $y = (2x - 1)/(2+x^2)$, [-2 ; 0]
- 4) $y = 2\sin 2x + 3\cos 2x$, [0 ; $\pi/4$]
- 5) $y = \operatorname{tg} 2x$, (-1 ; $\pi/2$)
- 6) $y = 3\sin x + 4\cos 3x$, [0 ;]
- 7) $y = (1 + x^2) \cos 2x$, [-1 ; 1]
- 8) $y = 2\sin 2x + 3\cos 2x$, [0 ;]

Quyidagi misollarda oraliqni mustaqil tanlab, shu oraliqda funksiya ekstremumlarini toping va turli oraliqlardagi natijalarni solishtirib, tahlil qiling:

- 1) $y = (1 + x^2) e^{-4x/3}$,
- 2) $y = x^2/\ln x$,
- 3) $y = \cos(\ln x)$
- 4) $y = (2+X^2)/\sin(x)$,
- 5) $y = \ln(1 + 2\cos x)$,
- 6) $z = x^2 - xy + y^2$
- 7) $z = y - y^2 - x + 6y$,
- 8) $z = x^3 + 8y^3 - 6xy + 1$,
- 9) $z = \sin x + \sin y + \sin(x+y)$,
- 10) $z = e^x/2(x + y^2)$,
- 11) $z = x \ln x$,
- 12) $z = (x+1)\operatorname{arctg} x$.

Nazorat savollari

1. Optimallashtirish masalasini keltiring.
2. Optimallashtirish usullarini aytib bering.
3. foptions massivining har bir komponentasini tushuntirib bering.
4. fminbnd funksiyasi qanday ko'rinishlarga ega?
5. fminsearch funksiyasi qanday ko'rinishlarga ega?
6. fminbnd va fminsearch funksiyalar har bir formatini tushuntirib bering.
7. Optimization kutubxonasining xususiyatlari qanday?

19. FUNKSIYA HOSILASINI CHEKLI AYIRMALAR BILAN APPROKSIMATSIYALASH VA SONLI INTEGRALLASH MASALALARI

19.1 Chekli ayirmalar

Funksiyalarning hosilalarini taqribiy hisoblash (sonli differensiallash) masalasini qarashdan avval chekli ayirmalarni amalga oshiruvchi Matlab funksiyalari bilan tanishib chiqaylik:

1. **diff(X)** - X massivning qo'shni elementlarini chekli ayirmalarini qaytaradi:

a) agar X vektor bo'lsa, **diff(X)** qo'shni elementlar ayirmalari vektori $[X(2)-X(1) \quad X(3)-X(2) \quad \dots \quad X(n)-X(n-1)]$ ni qaytaradi va uning elementlar soni X vektorga nisbatan 1 taga kam bo'ladi;

b) agar X matritsa bo'lsa, u xolda **diff(X)** ustunlar ayirmalari matritsasini beradi: $[X(2:m,:)-X(1:m-1,:)]$;

2. **diff(X,n)** - n-tartibli chekli ayirmalarni qaytaradi.

Masalan, $\text{diff}(X,2) = \text{diff}(\text{diff}(X))$ demakdir.

Hisoblashlarda quyidagi rekurrent formula qo'llaniladi:

$$\text{diff}(X,n) = \text{diff}(\text{diff}(X,n-1))$$

3. **Y=diff(X,n,dim)** funksiyasi matritsaning satrlar yoki ustunlar bo'yicha chekli ayirmalarini dim parametr qiymatiga boq'liq ravishda qaytaradi. Agar n tartib dim miqdorga teng bo'lsa yoki undan oshsa, u holda **diff(X)** bo'sh massivni qaytaradi.

Misollar:

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Program Files (x86)\RocketDock
Shortcuts How to Add What's New
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> X=[3 2 5 4 12 9 6]

X =
     3     2     5     4    12     9     6

>> size(X)

ans =
     1     7

>> Y=diff(X)

Y =
    -1     3    -1     8    -3    -3

>> Y=diff(X,2)

Y =
     4    -4     9    -11     0

>> X=magic(6)

```

19.1-rasm. Chekli ayirmalarni hosil qilish.

```

>> X=magic(6)

X =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> Y=diff(X,2)

Y =
    60   -54    -6     6     0    -6
   -51    42    36    -6   -21     0
    45   -42   -30     0    21     6
   -48    54    -6     6     0    -6

>> Y=diff(X,2,3)

Y =

Empty array: 6-by-6-by-0

```

19.1 - rasm. Matritsaning chekli ayirmasi.

19.2. Funksiya hosilasi

Funksiya hosilasini chekli ayirmalar bilan approksimatsiyalash uchun $\text{diff}(y)/\text{diff}(x)$ qoidadan foydalanamiz.

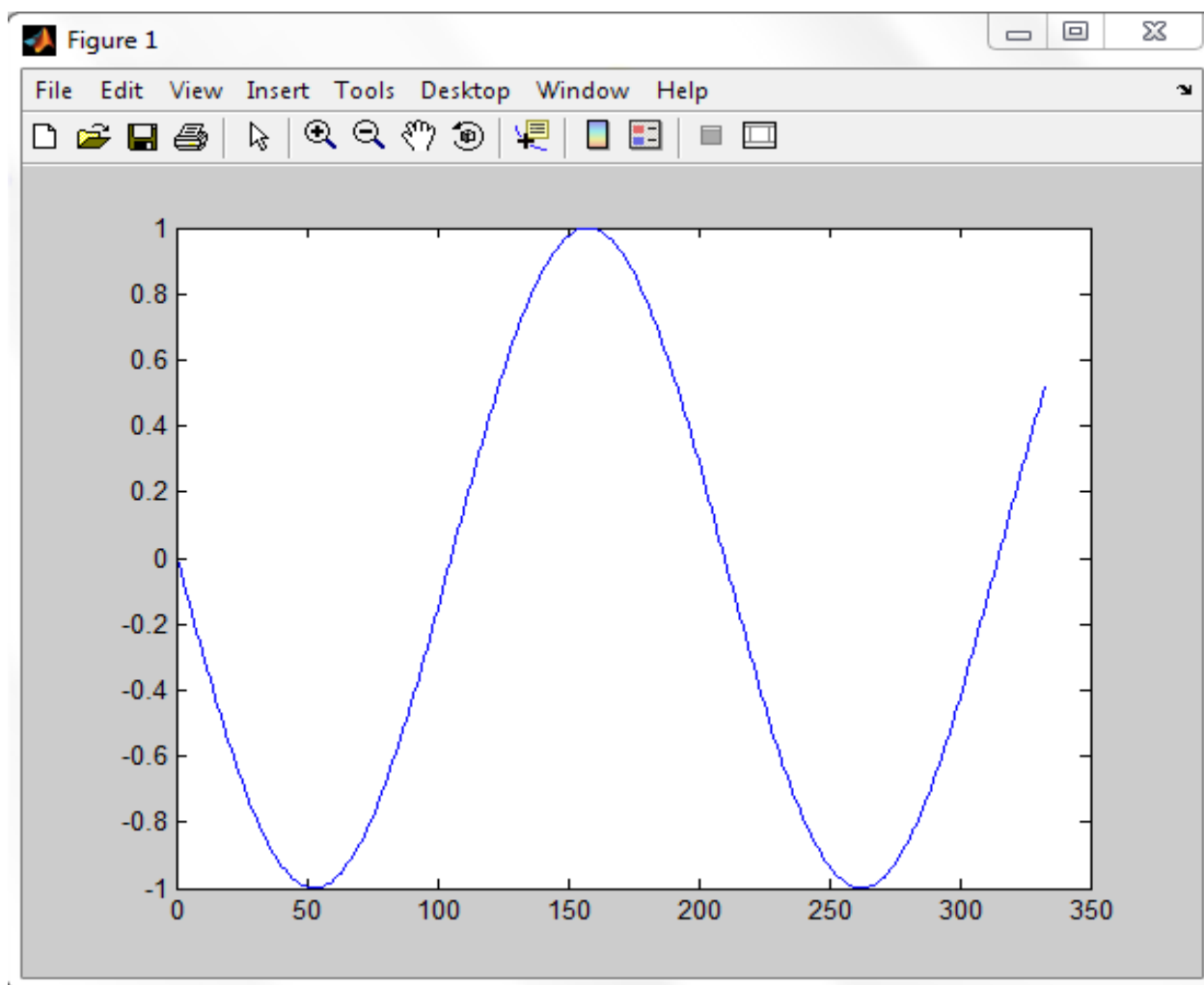
Funksiya hosilasini topish va hosila grafigini chizish masalasini quyidagi misol yordamida ko'rsak bo'ladi:

Misol:

```
>>h=0.05
```

```
>> X=0:h:10; S=cos(X);
```

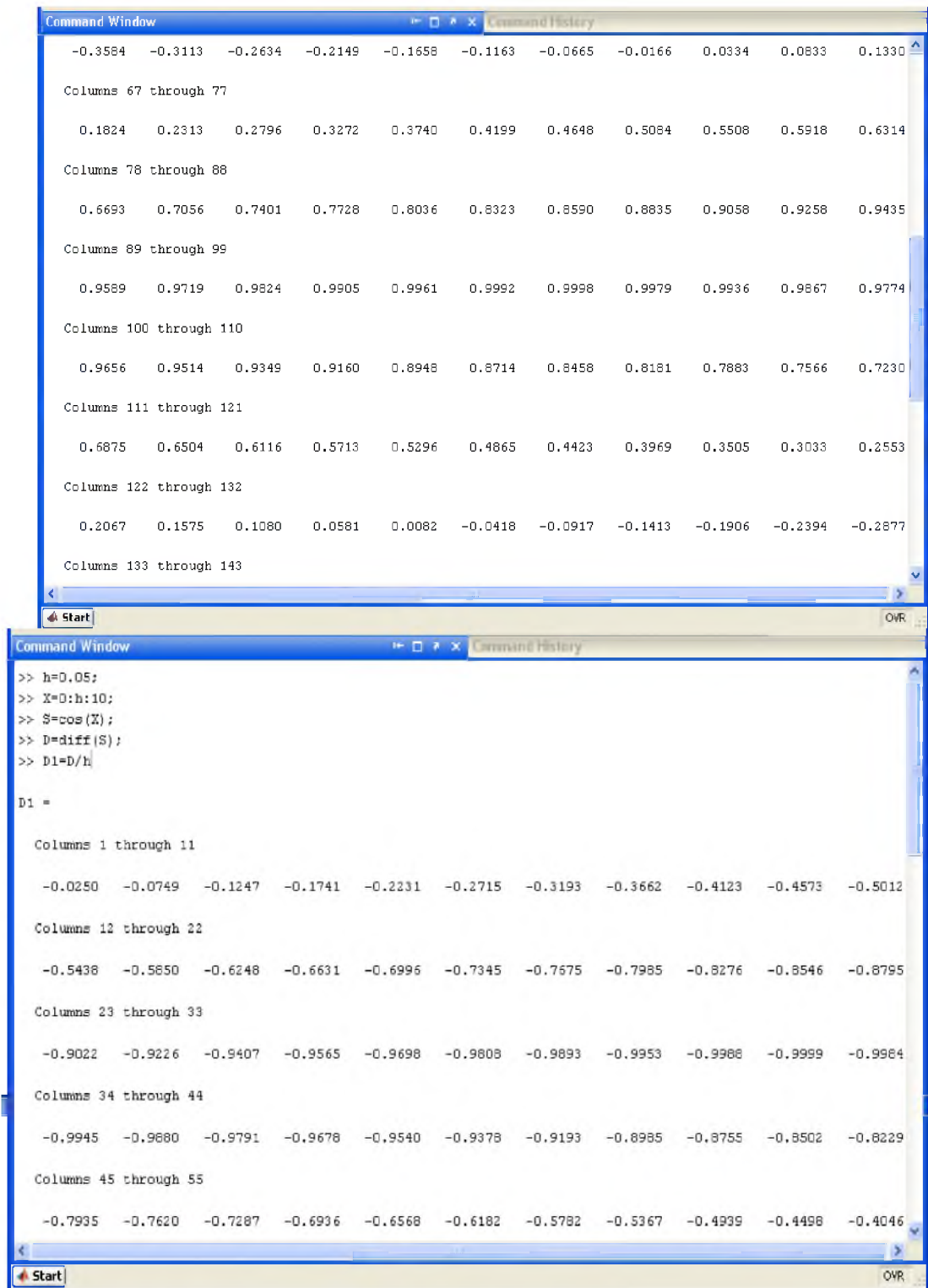
```
>> D=diff(S); plot(D/h)
```



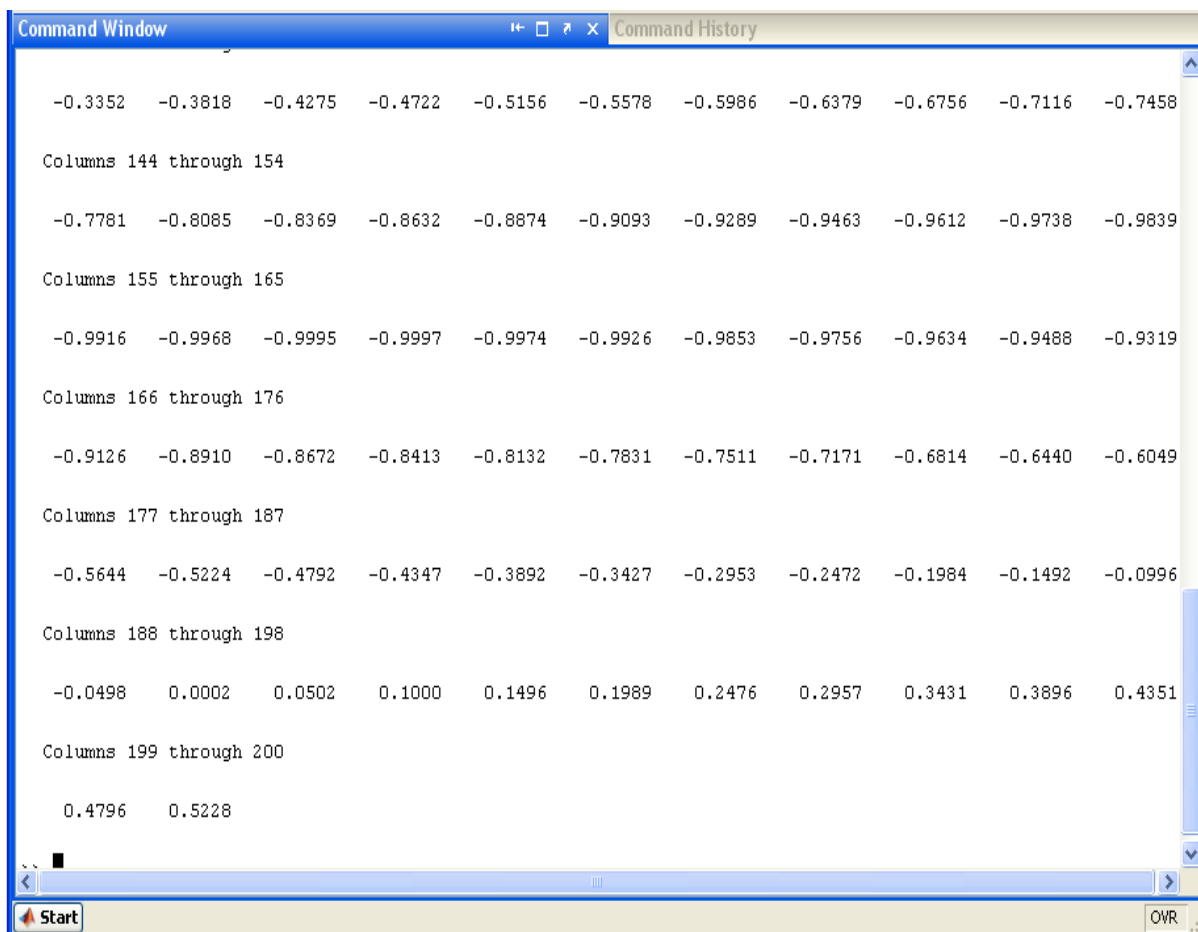
19.2 -rasm . $y=\cos(x)$ funksiya hosilasining grafigi.

E'tibor beraylik, \mathbf{x} o'qi bo'ylab \mathbf{X} vektorning elementlari qiymatlari emas, balki ularning tartib nomerlari berilgan.

Hosilaning chekli ayirmalar bilan approksimatsiyalash natijasida qabul qiladigan qiymatlarini ko'rish uchun qo'shimcha `>> D1=D/h` komandasini berish etarli:



19.3-rasm. Hosilaning approssimatsiyasi.



19.3 - rasm. Hosilaning approksimatsiyasi.

Symbolic Math Toolbox kengaytirilgan paketi funksiyaning analitik ko'rishda differensiallash imkonini ham beradi.

19.3. Funksiya gradientini hisoblash

Funksiya gradientini chekli ayirmalar usuli bilan hisoblash gradient funksiyasi orqali amalga oshiriladi. U quyidagi formatlarda qo'llaniladi:

1. $\mathbf{FX}=\mathbf{gradient}(F)$ - F vektor bilan berilgan bir o'zgaruvchili funksiya gradientini qaytaradi (hisoblaydi). FX- x yo'nalish bo'yicha chekli ayirmaga mos keladi;

2. $[\mathbf{FX},\mathbf{FY}]=\mathbf{gradient}(F)$ - FX va FY massivlar ko'rinishida F matritsa bilan berilgan ikki o'zgaruvchili $F(X,Y)$ funksiya gradientini qaytaradi. FX massiv x yo'nalish bo'yicha chekli ayirmaga(ustunlar), FY massiv esa y yo'nalish bo'yicha chekli ayirmaga (satrlar) mos keladi;

3. $[\mathbf{FX},\mathbf{FY},\mathbf{FZ},\dots]=\mathbf{gradient}(F)$ - ko'p o'lchamli F massiv ko'rinishida berilgan ko'p o'zgaruvchili funksiya gradientining qator komponentalarini qaytaradi;

4. $[...] = \text{gradient}(F, h)$ - har bir yo'nalish bo'yicha masofani tayinlash uchun h qadamdan foydalaniladi (h -skalyar miqdor);

5. $[...] = \text{gradient}(F, h_1, h_2, \dots)$ - agar F ko'p o'lchamli massiv bo'lsa, u holda masofa h_1, h_2, h_3, \dots parametrlar bilan aniqlanadi.

Misollarga murojaat qilaylik.

```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Program Files (x86)\RocketDock
Shortcuts How to Add What's New
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> F=[3 4 6 1 2 3 5 7 9]

F =

     3     4     6     1     2     3     5     7     9

>> FX=gradient(F)

FX =

     1.0000     1.5000    -1.5000    -2.0000     1.0000     1.5000     2.0000     2.0000     2.0000

>> F=[2 5 1 8 2; 2 45 2 32 2; 4 3 2 23 3; 8 7 12 21 3; 12 34 22 3 7]

F =

     2     5     1     8     2
     2    45     2    32     2
     4     3     2    23     3
     8     7    12    21     3
    12    34    22     3     7

>> [FX,FY]=gradient(F)

FX =

     3.0000    -0.5000     1.5000     0.5000    -6.0000
    43.0000         0    -6.5000         0   -30.0000
    -1.0000    -1.0000    10.0000     0.5000   -20.0000
    -1.0000     2.0000     7.0000    -4.5000   -18.0000
    22.0000     5.0000   -15.5000    -7.5000     4.0000

FY =

         0    40.0000     1.0000    24.0000         0
     1.0000    -1.0000     0.5000     7.5000     0.5000
     3.0000   -19.0000     5.0000    -5.5000     0.5000
     4.0000    15.5000    10.0000   -10.0000     2.0000
     4.0000    27.0000    10.0000   -18.0000     4.0000

```

19.4 –rasm. Funksiya gradienti.

Gradient funksiyasi ko'pincha gradientlar maydoni grafigini chizish uchun qo'llaniladi.

19.4. Sonli integrallash

Sonli integrallashda quyidagi aniq integral taqriban hisoblanadi:

$$\int_a^b y(x)dx \quad (1)$$

Aniq integral (1) ni taqribiy hisoblash usullaridan biri trapetsiya usuli bo'lib, uning Matlab tizimidagi foydalaniladigan funksiyalari quyidagicha formatlarda berilishi mumkin:

1. **trapz(Y)** - aniq integralni qaytaradi(integrallash qadami $h=1$). a) agar Y-vektor bo'lsa, trapz(Y)- Y ning elementlari integralini qaytaradi;b) agar Y matritsa bo'lsa, trapz(Y)- matritsa ustunlari integrallarini o'z ichiga oluvchi vektor -sitrni qaytaradi;

2. **trapz(X,Y)** -Y funksiyadan X o'zgaruvchi bo'yicha integralni qaytaradi (integrallash chegaralari X vektorning boshlang'ich va so'nggi elementlari yordamida beriladi);

3. **trapz(X,Y)**- o'zgaruvchining qiymatiga boq'liq holda matritsa uchun satrlar yoki ustunlar bo'yicha integralni qaytaradi.



```
MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Program Files (x86)\RocketDock
Shortcuts How to Add What's New
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> Y=[5 1 8 3 10 21]

Y =

     5     1     8     3    10    21

>> trapz(Y)

ans =

     35

>> X=0:pi/70:pi/2;
>> Y=cos(X);
>> Z=trapz(Y)

Z =

    22.2780
```

19.5-rasm. Integralning qiymati.

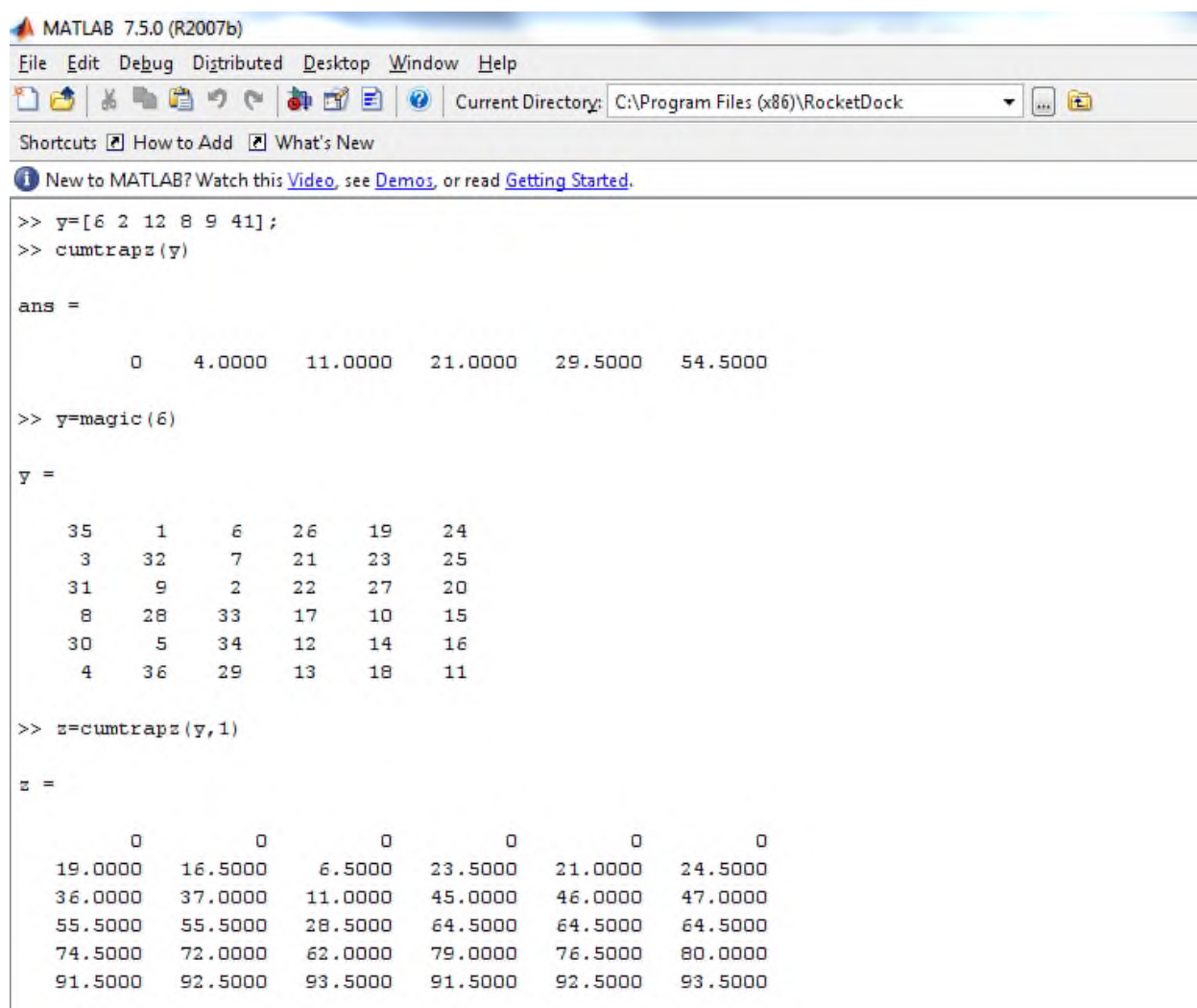
Quyidagi funksiyalardan foydalanganda integrallash to'planish bilan davom etadi:

1. **cumtrapz(Y)** -ordinatalari Y vektor (matritsa) ko'rinishida berilgan funksiyaning integrallash qadami $h=1$ bilan hisoblangan integralining son qiymatlarini qaytaradi. Agar qadam 1 dan farqli o'zgarmas bo'lsa, hisoblangan integralni qadamning kattaligiga

ko'paytirish yetarli. Ushbu funksiya vektorlar uchun vektorni, matritsalar uchun matritsani qaytaradi;

2. **cumtrapz(X,Y)** - X o'zgaruvchi bo'yicha Y integrallanadi. X va Y bir xil uzunlikdagi vektorlar yoki X vektor-ustun, Y esa matritsa bo'lishi kerak;

3. **cumtrapz(...,dim)** - dim skalyar bilan aniqlangan o'lcham bo'yicha integrallashni bajaradi.



```
MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Program Files (x86)\RocketDock
Shortcuts How to Add What's New
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> y=[6 2 12 8 9 41];
>> cumtrapz(y)

ans =

    0    4.0000   11.0000   21.0000   29.5000   54.5000

>> y=magic(6)

y =

    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> z=cumtrapz(y,1)

z =

     0     0     0     0     0     0
   19.0000   16.5000    6.5000   23.5000   21.0000   24.5000
   36.0000   37.0000   11.0000   45.0000   46.0000   47.0000
   55.5000   55.5000   28.5000   64.5000   64.5000   64.5000
   74.5000   72.0000   62.0000   79.0000   76.5000   80.0000
   91.5000   92.5000   93.5000   91.5000   92.5000   93.5000
```

19.6 - rasm. Integralni o'lchov bo'yicha hisoblash.

19.5. Kvadraturalar usulida integrallash

Trapetsiyalar usuli berilgan qadamlarda unchalik katta bo'lmagan aniqlikni ta'minlaydi yoki berilgan chegaradagi hisoblashlarni amalga oshirishda juda ko'p sondagi qadamlarni beradi.

Simpson kvadratur formulasi yoki Gaussa-Lobatto usulidan foydalangan holda integrallashni va ikki karrali integrallashni amalga

oshiruvchi quyidagi funksiya ancha aniq natija beradi (formulalardagi fun ifodasi odatda oddiy apostrof ichida yoki handle funksiya shaklida beriladi):

1. **quad(fun,a,b)** - [a b] kesmada fun funksiya aniq integralining sonli qiymatini qaytaradi.

2. **quad(fun,a,b,tol)**- berilgan tol nisbiy aniqlikda aniq integralning sonli qiymatini qaytaradi(sukut holatda $tol=1.e-6$).Shuningdek,nisbiy va absolyut aniqlik kombinatsiyasini aniqlash uchun ikki elementdan iborat $tol=[rel_tol\ abs_tol]$ vektordan ham foydalanish mumkin.

3. **quad(fun,a,b,tol,trace)** - aniq integralning sonli qiymatini qaytaradi va parametr trace nolga teng bo'lmaganda integralni hisoblash jarayonini ko'rsatuvchi grafik qyradi.

4. **quad(fun,a,b,tol,trace,P1,P2,...)** - fun funksiya aniq integralining sonli qiymatini qaytaradi, integral osti $G=fun(X,P1,P2,...)$ funksiya to'q'ridan-to'q'ri yuboriladigan $P1,P2,...$, qo'shimcha argumentlardan foydalaniladi.

Ikki karrali integrallarni hisoblash uchun Matlabda quyidagi funksiyalar mavjud:

1. **dblquad(fun, inmin, inmax, outmin, outmax)** - integral osti funksiyasi $fun(x,y)$ uchun ikki karrali integralni hisoblaydi, bu erda :

$inmin \leq x \leq inmax$ - ichki o'zgaruvchi-vektor ;

$outmin \leq y \leq outmax$ – tashqi o'zgaruvchi-skalyar.

2. **dblquad(fun, inmin, inmax, outmin, outmax,tol,trace)**- dblquad funksiyaga tol, trace parametrlarni beradi va h.k.

Ma'lumotnomaga murojat qilib , quad funksiyasi haqida qo'shimcha axborotlarni olish mumkin.

Nazorat savollari

1. Qaysi funksiya chekli ayirmani amalga oshiradi?
2. Agar o'zgaruvchi massiv bo'lsa, chekli ayirma qanday hisoblanadi?
3. Chekli ayirmalar uchun qanday rekurrent formula mavjud?
4. Satrlar bo'yicha chekli ayirmalar qanday topiladi?
5. Hosilaning son qiymatini topish uchun qanday yo'l tutish kerak?
6. Funksiya gradientini hisoblovchi Matlab funksiyasini ayting.
7. Matlabda sonli integrallash uchun qaysi usullar qaralgan?
8. Qaysi funksiya trapetsiya usuliga asoslanadi?

9. Kvadraturalar bo'yicha integrallashni qanday Matlab funksiyasi amalga oshiradi?

10. Ikki karrali integralni hisoblovchi Matlab funksiyasi formatlarini tushuntiring.

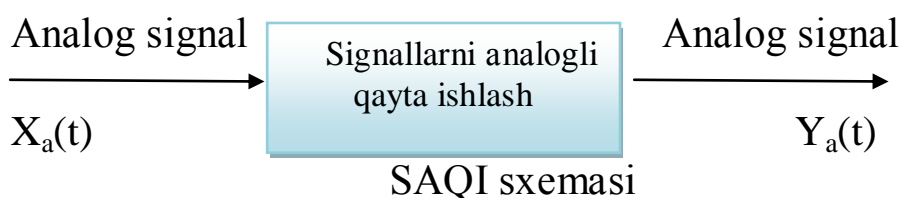
20. MATLAB FUNKSIYALARINI SIGNALLARNI RAQAMLI QAYTA ISHLASH MASALALARIGA QO'LLANILISHI

20.1. Signallarni raqamli qayta ishlash tushunchasi (SRQI)

Zamonaviy dunyoda odamlar har xil ko'rinishdagi signal to'rlari bilan o'ralgan. Ulardan ayrimlari tabiiy bo'lsa, asosiy qismi insonlar tomonidan hosil qilingan. Ayrimlari zarur signallar (nutq), ayrimlari yoqimli (muzika). Injenerlar nuqtai-nazaridan signallar- axborot tashuvchilardir. Demak, foydali axborotni qarama-qarshi axborot "aralashmasi" dan olib tashlash yoki kuchaytirish- signalni qayta ishlashni oddiy ko'rinishidir.

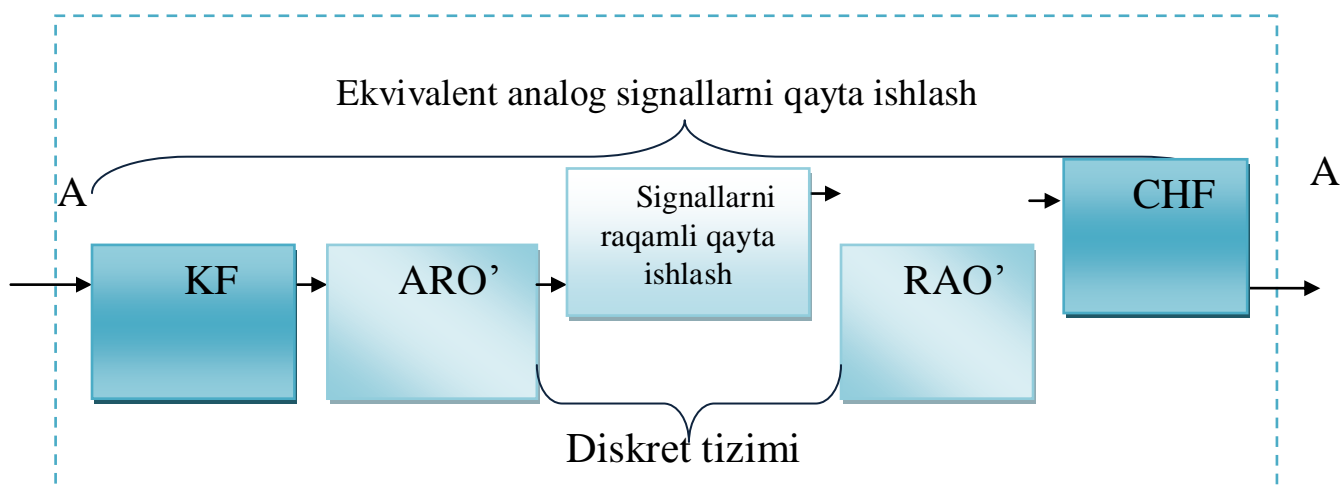
Umuman olganda, axborotni qayta ishlash deganda axborotni qabul qilish, kengaytirish, yaxshilash, saqlash va kerakli ma'lumotni uzatish jarayoni tushiniladi.

Quyida SAQI (signallarni analogli qayta ishlash) qanday bo'lishini ko'ramiz. Amaliyotda duch keladigan signallarning asosiy qismi analog signallardir. Bu signallar vaqt va amplituda bo'yicha uzluksiz o'zgarib turadi va aktiv-passiv elementli sxema yordamida qayta ishlanadi. Bu munosabat bizga SAQI kabi ma'lum, masalan, radiopriyomnik va televizor.



Bu signallarni summator, kuchaytirgich va mantiqiy elementdan iborat bo'lgan raqamli apparat vositalari yoki maxsus vazifalarni bajaruvchi mikroprosessorlar yordamida qayta ishlanadi.

Biroq analog signallar ko'rinishini o'zgartirish raqamli apparat ta'minotiga mos kelishi talab qilinadi. Bu signal ko'rinishi raqamli signal deyiladi. Signal vaqtning aniq momentda ularning sonidan oxirgi bitta qiymatni qabul qiladi va haqiqatdan ham ikkilik raqam yoki bitlarda ko'rsatish mumkin.



ARO'- analog- raqamli o'zgartirgich, analog signaldan bitlar oqimini tashkil qiladi;

RAO' –ikkilik sonlar ketma-ketligidan zinapoyali to'lqinlar hosil qiluvchi raqamli-analogli o'zgartirgich;

ChF- zinapoyali to'lqinlarni kerakli analogli signalga silliqlovchi chiqish filtri;

Signallarni raqamli qayta ishlash- SRQI ni “yuragi”, umumiy ishlarga mo'ljallangan kompyuter, maxsus vazifa uchun ishlovchi prosessor, raqamli sxema va boshqalardan iborat bo'lishi mumkin .

SAQI ning eng katta kamchiligi signalni murakkab ilovalarda qayta ishlash imkoniyatining chegaralanganligidir. Bu esa tizimni loyihalashtirishning murakkabligi va signallarni qayta ishlashdagi moslashuvchanlikda aks etadi. Natijada, bu yakuniy mahsulot va ilovalarni qimmatlashishiga olib keladi. Boshqa tomondan SRQI dan foydalanib, uncha qimmat bo'lmagan ShK signallarni qayta ishlovchi kuchli kompyuterga aylantirsa bo'ladi.

Yuqorida ko'rsatilgan holda signallarni qayta ishlash shu narsaga olib kelishi mumkinki, SRQI juda murakkab va ,“codda ko'ringan” SAQI ga nisbatan, ko'p komponentlardan tashkil topgan. Shunga qaramasdan SRQI da bir nechta afzalliklar bor. Bunday afzalliklarga quyidagilar kiradi:

1.SRQI dan foydalanuvchi tizimlar kompyuterda ishlaydigan dasturiy ta'minotdan foydalanib ishlab chiqilishi mumkin.Shuning uchun SRQI ni testlash va ishlab chiqish juda qulaydir.

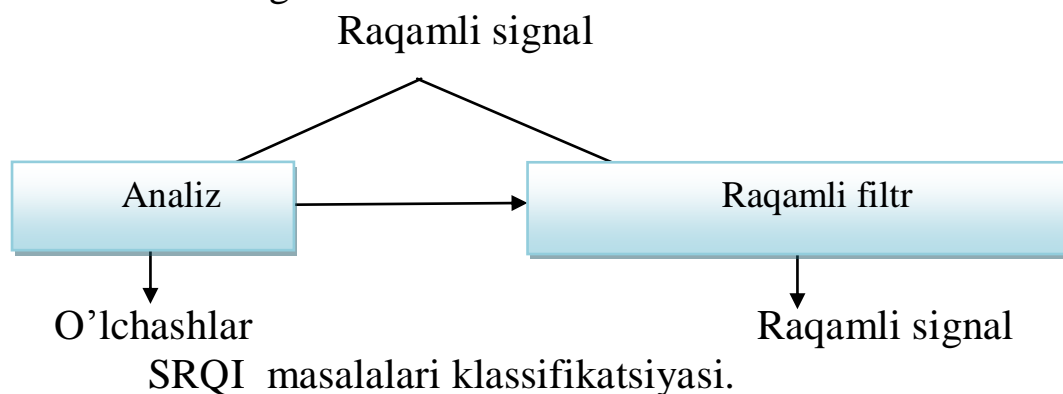
2.SRQI da operatsiyalar faqat qo'shish va ko'paytirishga asoslangan. Bu esa qayta ishlashni turq'unligini ta'minlab beradi;

3. Dasturga o'zgartirish kiritib, SRQI operatsiyalarini oson modifikatsiyalash mumkin;

4. SRQI ning arzonligi.

SRQI ning yuqorida aytib o'tilgan afzalliklari undan ko'pgina texnologiya va ilovalarni yechishda foydalanishga asos bo'ladi, masalan, maishiy elektronika, kommunikatsiya, mobil aloqa.

SRQI ning ko'p operatsiyalari quyidagicha: signallarni tahlil qilish yoki filtrlash masalalariga bo'linishi mumkin.



20.2. Signallarni tahlil qilish va filtrlash

Signallarni tahlil qilish masalasi signal xossalarini o'zgarishiga boq'liq. Ba'zi ilovalarda bulardan quyidagilar zarur:

- spektral (chastotali yoki ikki fazali) analiz tahlili;
- nutqni tanish;
- maqsadlarni aniqlash.

Signallarni filtrlash masalalari kirishdagi signal-chiqishdagi signal holati bilan xarakterlanadi. Tizimlarda bu masalalarni bajaruvchi qism umumiy holda filtrlar deb ataladi. Odatda ular vaqt sohasidagi operatsiyalar bo'ladi.

Quyidagi filtratsiya ilovalari mavjud:

- istalmagan fonli shovqinni o'chirish;
- shovqinlarni o'chirish;
- chastotali polosalarni ajratish;
- signal spektrlarini hosil qilish.

Ba'zi ilovalarda signal avval uning xarakteristikasini o'rganish uchun tahlil qilinadi. Ulardan sintetik ovozni amalga oshirish uchun raqamli filtratsiyada foydalaniladi.

20.3. SRQI masalalarini echish uchun MATLAB muhiti

MATLAB yadrosining o'zi SRQI ni amalga oshirish uchun hech qanday maxsus vositaga ega emas. Shuning uchun foydalanuvchilar shunday vositalarga qiziqqanda, ularni o'zlari ishlab chiqarishlari kerak bo'ladi. Yuqorida aytilganidek, MATLAB signallarni qayta ishlash uchun ikkita kutubxonaga ega: Signal Processing va Wavelet. Bu kutubxonalarda foydalanish uchun juda ko'p funksiyalar bor. Shuning uchun istalgan foydalanuvchi amaliy jihatdan ulardan foydalanishi mumkin.

Signal Processing kutubxonasi bevosita signallarni qayta ishlashga mo'ljallangan. Bunda 100 dan ortiq funksiya bor. Hamma funksiyalar ma'nosi bo'yicha guruhlab joylashtirilgan. Ular quyidagilar:

-“Signallarni amalga oshirish va ularni grafik tasvirlash”. Berilgan formadagi (sinusoidal, arrasimon, to'q'ri burchakli impulslar va boshqa) signallarni hosil qilish uchun mo'ljallangan funksiyalardan tashkil topgan;

-“Filtrlarni tahlil qilish va ishlatish”. Bu bo'lim funksiyalari filtrlashning ba'zi standart algoritmlarini amalga oshiradi. Boshlang'ich ma'lumotlar(kiruvchi signal), shuningdek filtr parametrlarini funksiya ko'rinishida ishlatib, mos funksiyalarni chaqirganda uzatiladi;

-“Tizimlarni chiziqli almashtirish”-polinom ko'rinishida berilgan tizimlarni biridan boshqasigi o'zgartiruvchi funksiyalar;

-“Cheksiz impuls harakterisikali(ChZIX) to'q'ri va klassik filtrni ishlab chiqish”- filtrlarning ba'zi klassik modellarini ishlab chiqish imkonini beradi (masalan, Bassel, Chabishev va Battervort filtrlari).

-“ChIX dan filtr tartibini tanlash”- Battervort, Chebishev va elliptik filtrlarni tartibini tanlash funksiyasidan tashkil topgan;

-“Chekli impuls xarakteristikali filtrni ishlab chiqish(ChKIX)”- ChKIX filtrlarini loyihalash uchun oynalar, kichik kvadratlar usuli va boshqa standart metodlardan foydalanuvchi funksiyalardan tarkib topgan;

-“Almashtirish”- Fur'ening to'q'ri va teskari almashtirishini hamda Gilbert va Z-almashtirishlarni amalga oshiruvchi funksiyalardan tashkil topgan;

-“Signallarni statistik qayta ishlash ”- signallarni statistik qayta ishlashni bajaradi, ba'zi statistik parametrlarni aniqlaydi;

-“Oyna”- turli oynalar metodini amalga oshiradi (Bartleta, Chebisheva, Kayzer oynasi va b);

-“Parametrik modellashtirish”-ma'lumotlar asosida filtrlarni identifikatsiya qilishga ruhsat beradi;

-“Maxsuslashtirilgan operatsiyalar”-ma'lumotlar ustida qo'shimcha operatsiyalarni bajarishga mo'ljallangan funksiyalardan tarkib topgan;

-“Analogli prototipni ishlab chiqish”- klassik filtrlarni analogli prototipini ishlab chiquvchi funksiyalar ;

-“Chastotani o'zgartirish”-past chastotali analogli signalni boshqasiga o'zgartiruvchi funksiyalardan tashkil topgan;

-“Filtrni diskretlash”- filtrlarni analoglidan raqamligiga o'zgartirish funksiyalari;

-“Interaktiv vositalar”- sptool funksiyasiga ega bo'lib , signallarni qayta ishlash uchun interaktiv vizual vositalarni yuklaydi.

20.4. SRQI standart masalalarini yechishga doir misol

SRQI standart masalalaridan birini ko'ramiz. Signal spektrini aniqlash uchun Fur'e almashtirishidan foydalaniladi. Bu masalalarni yechish uchun avval signal spektri tushunchasini aniqlashimiz kerak. Agar qandaydir tebranish jarayoni turli chastotalarni garmonik tebranishlari yig'indisi ko'rinishida ifodalansa, unda turli chastota bo'yicha amplituda taqsimotini ifodalovchi funksiya tebranish jarayonining spektri deyiladi.

Signal spektrini aniqlash uchun Furening to'fri va teskari almashtirishlaridan foydalaniladi, yani chastotali sohada signalni tavsiflash uchun qo'llaniladi.

Analogli signal $x_a(t)$ ning spektri $X_a(j\omega)$ deb to'q'ri Fure almashtirishiga aytiladi:

$$X_a(j\omega) = \int_0^{\infty} X_a(t)e^{-j\omega t} dt \quad (1)$$

Teskari Fure almashtirishi yordamida spektr orqali signalni aniqlab olish mumkin.

$$X_a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_a(j\omega)e^{j\omega t} d\omega \quad (2)$$

Diskret signal $x(nT)$ ning spektri $X_a(j\omega T)$ deb to'q'ri Fur'e almashtirishiga aytiladi:

$$X(e^{j\omega T}) = F\{x(nT)\} = \sum_{n=0}^{\infty} x(nT) e^{-j\omega nT} \quad (3)$$

$x(nT)$ signal spektr orqali teskari Fur'e almashtirishi yordamida aniqlanadi.

$$x(nT) = \Phi^{-1}\{X(e^{j\omega T})\} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} X(e^{j\omega T}) d\omega \quad (4)$$

Fur'ening diskret almashtirishi (FDA) deb quyidagicha o'zaro bir qiymatli almashtirishga aytiladi:

$$X(k) = X(k\Omega) = \sum_{n=0}^{N-1} x(nT) e^{-jkn\Omega T}, \quad k=0,1,\dots,N-1, \quad (5)$$

$$x(nT) = \frac{1}{N} \sum_{k=0}^{N-1} X(k\Omega) e^{jkn\Omega T}, \quad n=0,1,\dots,N-1, \quad (6)$$

bu yerda $x(nT)$ davriy ketma-ketlik bo'lib, davri- NT .

(5) tenglik Fur'ening to'g'ri diskret almashtirishi deyiladi, (6) esa teskarisi deyiladi (FTDA).

Bu almashtirishlarda $\Omega = \frac{2\pi}{NT}$ FDA ning asosiy chastotasidir. Yuqorida $e^{-j\Omega T} = e^{-j2\pi/N} = W_N$, belgilash kiritib, Fur'ening diskret almashtirishlarini quyidagicha yozish mumkin:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0,1,\dots,N-1 \quad (7)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad n = 0,1,\dots,N-1 \quad (8)$$

Bunda $X(k)$ huddi $x(n)$ ketma-ketlikning o'zi kabi k bo'yicha N davrli davriy funksiyadir. Chunki, m -butun bo'lganda

$$W_N^{kn} = W_N^{(k+mN)n}$$

bo'ladi. Diskret Fur'e o'zgaruvchisi chekli N uzunlikdagi $x(nT)$ ketma-ketlikni tasvirlash uchun foydalaniladi ($n=0,1,2,3,\dots,N-1$, $[0;N-1]$ intervaldan tashqarida esa u nolga teng). Haqiqatdan ham bunday ketma-ketlikni bitta davriy ketma-ketlikning bitta davri deb qarab, (7), (8) almashtirishlardan foydalanish mumkin ($[0; N-1]$ intervaldan tashqarida $X(k)$ va $x(n)$ larni nolga teng deb hisoblanadi).

Signal Processing kutubxonasi FDA ni bajarish uchun ikkita funksiyaga ega:

1) $y = \text{fft}(x, N)$ - N nuqtali FDA ni hisoblaydi. Agar x vektor uzunligi N dan kichik bo'lsa, x nolga to'ldiriladi. Agar N argument yozilmay qoldirilgan bo'lsa, FDA ning uzunligi x vektor uzunligiga teng deb hisoblanadi. Agar x matritsa bo'lsa, N nuqtali FDA x ning har bir ustuni uchun bajariladi.

2) $y = \text{ifft}(x, N)$ - N nuqtali FTDA ni hisoblaydi. Bu funksiya parametrlari ham yuqoridagi kabidir.

Bu funksiyalarning o'ziga xosligi ularni mashina tilida yozilganligidir. $\text{fft}()$ funksiyadan foydalanuvchi quyidagi misolni ko'ramiz: signal $s = 3.5 \cos(0.3\pi t)$ ifoda bilan tavsiflansin. Diskretlash

vaqti $T_s=0.3$ sek, $N =30$.Signal ustida FDA bajaramiz va berilgan vaqt kesmasida signal grafigini quramiz.

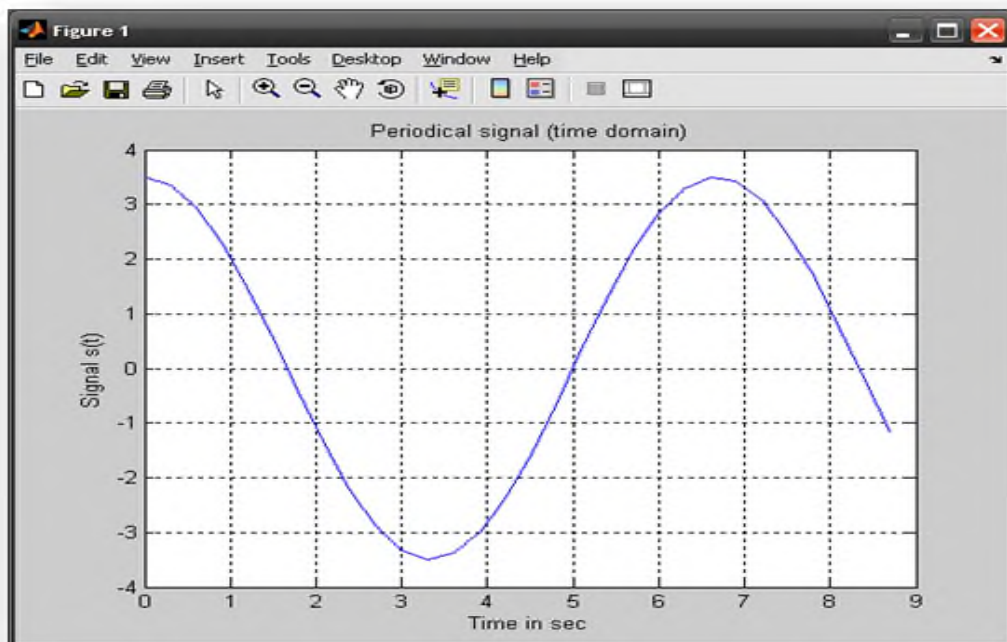
```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> T_s=0.3;
>> N=30;
>> time=[0:N-1]*T_s;
>> S=3.5*cos(0.3*pi.*time);
>> figure
>> plot(time,S);
>> grid
>> xlabel('Time in sec')
>> ylabel('Signal s(t)')
>> title('Periodical signal (time domain)')
>> zoom
>> S_FT=fft(S);
>> freq_plot=[0:N-1]./(N-1);
>> figure
>> zoom
>> plot(freq_plot,real(S_FT)),grid
>> title('Absolute value of transformed signal(frequency domain)');
>> ylabel('Abs. val. of signal')
>> xlabel('Frequency(in pi units)')
>> figure
>> zoom
>> plot(freq_plot,imag(S_FT)),grid
>> title('Phase (angle) of transformed signal(frequency domain)');
>> ylabel('Phase of signal')
>> xlabel('Frequency(in pi units)')
>> |

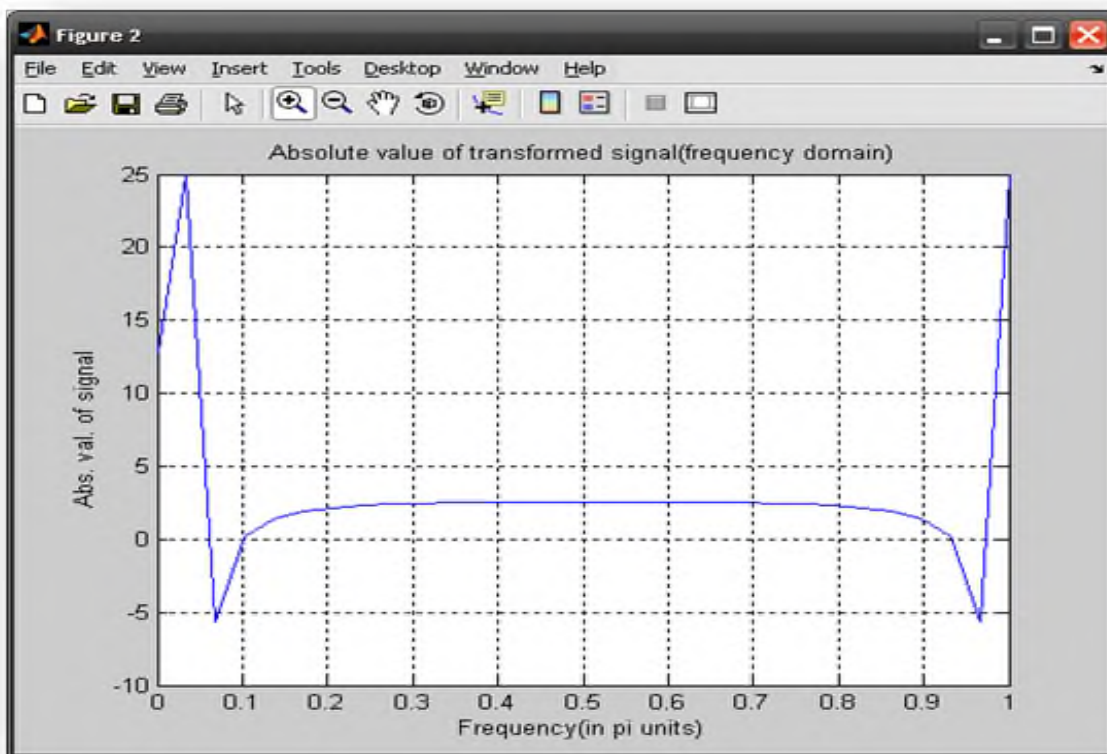
```

20.1- rasm. Signal grafigini qurish.

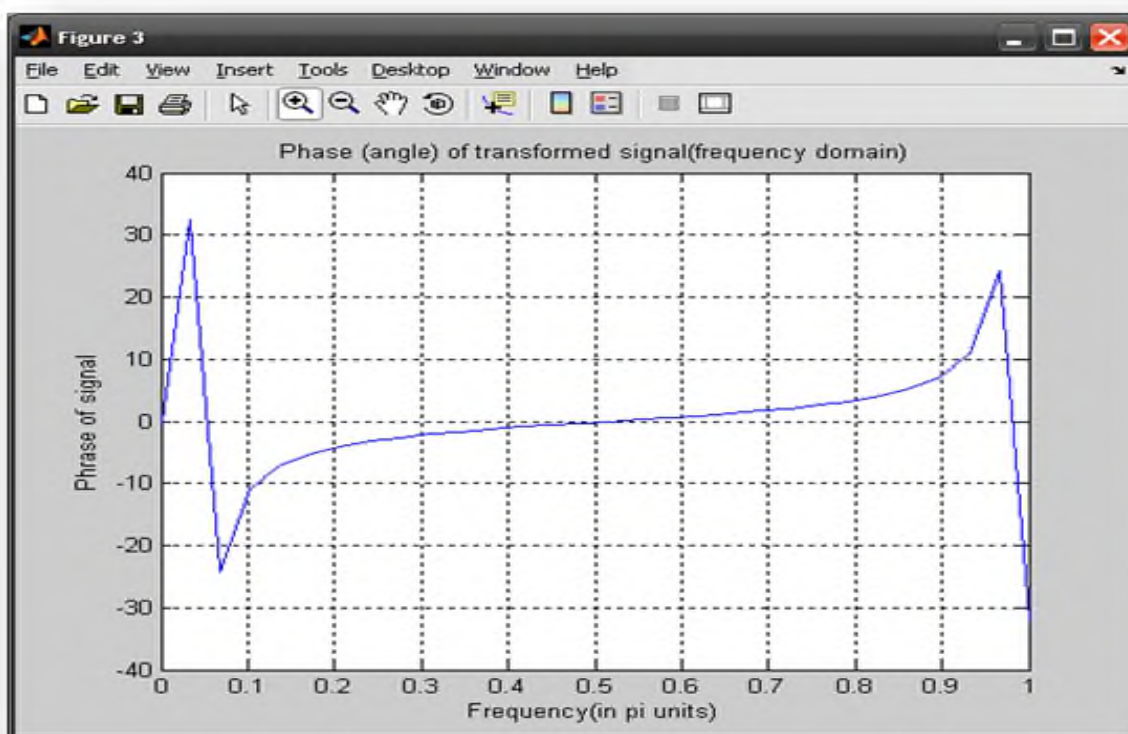
Bu dasturning natijalari quyidagi rasmlarda ko'rsatilgan:



20.2 - rasm. Berilgan vaqt kesmasida signal grafigi.



20.3 - rasm. Chastota sohasidagi signal amplituda grafigi.



20.4-rasm. Chastota sohasidagi signal faza grafigi.

Nazorat savollari

1. SRQI blok-sxemasini chizing.
2. Signallarni analog qayta ishlash va signallarni raqamli qayta ishlash bir-biridan qanday farq qiladi?
3. SRQI ning afzallik va kamchiliklarini ayting.
4. Signal Processing kutubxonasida nima ishlar bajariladi?
5. Spektr bu nima?
6. Fur'e o'zgaruvchisi nima? U nima uchun xizmat qiladi?

21. MATLAB YORDAMIDA DIFFERENSIAL TENGLAMALARNI YECHISH

21.1. Differensial tenglamalarning matematik tavsifi

Ko'plab tabiiy jarayonlar, chiziqli va chiziqsiz dinamik tizimlar va qurilmalarning matematik modellari differensial tenglamalar sistemasi (DTS) dan iboratdir. Shuning uchun DTS ni o'rganish va yechish alohida ah amiyat kasb etadi.

1-ta'rif. Differensial tenglama (DT) deb erkin o'zgaruvchi t , no'malum funksiya $y=y(t)$ va uning hosilalarini bog'lovchi tenglamaga aytiladi.

Agar noma'lum funksiya bir o'zgaruvchili (ko'p o'zgaruvchili) bo'lsa, tenglama oddiy (xususiy hosilali) differensial tenglama deyiladi.

Differensial tenglamaning tartibi deb unda qatnashayotgan hosilalarning eng katta tartibiga aytiladi.

Oddiy differensial tenglama (ODT) larni umumiy holda (oshkormas)

$$F(t, y, y', \dots, y^{(n)}) = 0,$$

xususan, 1-tartibli ODT ni

$$F(t, y, y') = 0, \quad (1)$$

ko'rinishida ifodalash mumkin.

Agar (1) tenglamani hosilaga nisbatan yechish mumkin bo'lsa, u holda ushbu oshkor ko'rinishdagi tenglamaga ega bo'lamiz:

$$y' = f(t, y) \quad (2)$$

2-ta'rif. (2) tenglamaning yechimi deb uni ayniyatga aylantiruvchi $y=\varphi(t)$ funksiyaga aytiladi, bu yechimning grafigi esa integral egri chiziq deyiladi.

Berilgan (2) tenglamaning umumiy echimi deb, s o'zgarmasning ixtiyoriy qiymatida uni qanoatlantiruvchi $y=\varphi(t, s)$ funksiyaga aytiladi. s o'zgarmasning biror s_0 qiymatida $y=\varphi(t, s_0)$ funksiya (2) tenglamaning xususiy yechimi deyiladi.

Yuqoridagi (2) tenglamaning umumiy yechimida ishtirok etuvchi S o'zgarmas odatda "boshlanq'ich" deb ataluvchi shartlar (**Koshi** shartlari) asosida aniqlanadi.

Koshi masalasi: (2) tenglamaning $y_{t=t_0}=y_0$ boshlang'ich shartni qanoatlantiruvchi yechimi aniqlansin.

Umuman olganda, (1) tenglamani (2) ko'rinishga analitik usulda keltirish har doim ham mumkin emas. Garchi, oshkormas DT ni yechimini analitik usulda topish murakkab masalalardan hisoblansada, ularni sonli

usullar yordamida taqribiy yechimlarini aniqlash muammo tug'dirmaydi. Sonli usullar taqribiy yechimni jadval ko'rinishda beradi.

DT ni sonli usul bilan echish deganda t argumentning berilgan $t_0, t_1, t_2, \dots, t_n$ qiymatlar ketma-ketligi va u_0 uchun $y = F(t)$ funksiyani aniqlamagan holda, u funksiyaning

$$y_i = F(t_i), i=1, 2, n,$$

$$y_0 = F(t_0)$$

shartlarni qanoatlantiruvchi $u_0, u_1, u_2, \dots, u_n$ qiymatlarini topish tushuniladi. Ushbu $h = t_k - t_{k-1}$, miqdor integrallash qadami deyiladi.

Sonli usullarni 2 guruhga ajratish mumkin:

1. Bir qadamli – bunda egri chiziqning bitta nuqtasi haqidagi axborot ishlatiladi va iteratsiya amalga oshirilmaydi (bunda echimni aniqlashni boshlash va h ni o'zgartirish mumkin, lekin funksiya qiymatlari ko'p martalab hisoblanadi). Mashina vaqti ko'p sarflanadi). Misol: Runge-Kutta, Eyley usullari.

2. Ko'p qadamli – bu holda egri chiziqning navbatdagi nuqtasini funksiya qiymatlarini takror-takror hisoblamasdan ham aniqlash mumkin (echishni boshlash mumkin emas, h o'zgartirilsa, bir qadamli usullarga qaytish kerak, ammo mashina vaqti tejaladi, cheklanish xatoligi haqida axborotni olish mumkin). Ko'rinib turibdiki, bu ikkala usulni birgalikda ishlatish yaxshiroq natija beradi.

21.2. Oddiy differentsial tenglamalar sistemasi(ODTS)ni yechish uchun MATLAB “Yechgich” lari

Matlabda ODTS ni yechish uchun quyidagi usullar (“yechgichlar”) taklif etiladi (ode – original differential equations):

1. **ode 45** – bir qadamli 4 va 5- tartibli Runge-Kutta oshkor usullari (u qattiq bo'lmagan tenglamalar sistemasini yechishda yaxshi natijalar beradi);

2. **ode 23** – bir qadamli 2- va 4- tartibli Runge-Kutta oshkor usullari (mo'tadil qattiq ODTS uchun yechim aniqlik darajasi ahamiyatsiz bo'lgan holda);

3. **ode 113** – ko'p qadamli o'zgaruvchi tartibli Adams-Bashvoriy-Multon usuli (bu yuqori darajada aniqlikni taminlovchi adaptiv usul);

4. **ode 15S** – sonli differentsiallash formulalaridan foydalanuvchi ko'p qadamli o'zgaruvchi tartibli (1 dan 5 gacha, avtomatik xolda – 5) usul (bu adaptiv usulni ode 45 “yechgich” echa olmagan holda yoki ODTS qattiq bo'lgan hollarda qo'llash maqsadga muvofiq);

5. **ode 23S** – bir qadamli (ODTS larning qattiq sistemalarini pastroq aniqlik bilan yechishda yuqori tezlikni taminlaydi) ;

6. **ode 23t** – interpolyatsiyali trapetsiya usuli (bu usul garmonik chiqish signalni tebranuvchi sistemalarni xarakterlovchi ODTS ni va mo'tadil qattiq ODTS ni echishda yaxshi natijalar beradi) ;

7. **ode 23tb** – dastlab (yechish boshlanishida) Runge-Kutta oshkormas usulini va keyin 2-tartibli teskari differensiallash formulasini qo'llovchi usul (aniqlik darajasi pastroq bo'lsa-da, ode 15s dan ko'ra samaraliroq);

8. **bvp 4c** – bu usul $y' = f(y, t)$, $F(y(a), y(b), p) = 0$ ko'rinishdagi ODTS uchun chegaraviy masalani yechishda qo'llaniladi.

Bu keltirilgan “yechgich” lardan quyidagicha foydalanish tavsiya etiladi:

- yuqorida keltirilgan barcha “yechgich” lar yordamida $y' = f(y, t)$ ko'rinishdagi ixtiyoriy ODTS ni yechish mumkin;

- ODT larning qattiq sistemasini yechish uchun esa ode 15s, ode 23s, ode 23t, ode 23tb lardan foydalanish mumkin ;

- ode 15s va ode 23t “yechgich” lar ushbu $M(t)y' = f(y, t)$ differensial-algebraik tenglamani yechadi ($M(t)$ – massa matritsasi deyiladi) ;

- ode 15s, ode 23s, ode 23t va ode 23tb lar ushbu

$$M(t, y)y' = f(y, t) \quad (3)$$

oshkormas tenglamani yechadi;

- barcha “yechgich” lar (3) ko'rinishdagi matritsaviy tenglamani yechishi mumkin (ode 23s va bvp 4c dan tashqari).

21.3. Differensial tenglamalarni yechish uchun funksiyalar

Bu “yechgich” lardan foydalanish uchun Matlabda quyidagi funksiya formatlari mavjud (“solver” o'rnida ixtiyoriy “yechgich” nomi bo'lishi mumkin);

1. **[T, Y]= solver('F', tspan, y₀);**

2. **[T, Y]= solver('F', tspan, y₀, options);**

3. **[T, Y]= solver('F', tspan, y₀, options, p₁, p₂, ...);**

4. **[T, Y, TE, YE, IE]= solver('F', tspan, y₀, options);**

5. **[T, Y]= solver('model', tspan, y₀, options, ut, p₁, p₂, ...)**

Bu erda :

1) F – odefile nomi, vektor-ustunni qaytaruvchi t va u ning funksiyasi;

2) $tspan$ – integrallash intervali $[t_0 \ t_{final}]$ ni aniqlovchi vektor. Vaqtning ma'lum o'sish yoki kamayish tartibida berilgan $t_0, t_1, \dots, t_{final}$ momentlarida echimni olish uchun $tspan = [t_0, t_1, \dots, t_{final}]$ komandani ishlatish kerak;

3) y_0 – boshlang'ich shartlar vektori;

4) $options$ – odeset (odeget yoki bvpget (faqat bvp4s)) funksiyasi hosil qilgan qo'shimcha argumentlarni, parametrlarni chiqarishga yordam beradi;

5) p_1, p_2, \dots – F funksiyaga taqdim etiladigan qo'shimcha parametrlar;

6) T, Y – yechimlar matritsasi Y , bunda har bir satr T vektor-ustun qaytaradigan vaqtga mos keladi;

Bu formatlarning mazmuni bilan alohida tanishaylik:

- $[T, Y]= solver ('F', tspan, y_0)$; - $y'=F(t, y)$ ko'rinishdagi sistemani $tspan [t_0 \ t_{final}]$ oraliqda y_0 boshlanq'ich shartlar asosida integrallaydi; ' F ' – ode file nomi (ODE funksiya deskriptori $@F$ ko'rinishda bersa ham bo'ladi) massivlarning har bir satri T vektor-ustundagi vaqtning aniq qiymatiga mos;

- $[T, Y]= solver ('F', tspan, y_0, options)$ huddi yuqoridagi format kabi, faqat integrallash parametrlari $options$ -argumentda ko'rsatilgan xossalarga ega bo'ladi ($options$ – argumentni odeset funksiyasi hosil qiladi). Odatda ishlatilayotgan parametrlar skalyar nisbiy xatolik $RelTol$ ni (avtomatik ravishda $1e-3$) va absolyut xatoliklar vektori $AbcTol$ ni (avtomatik tarzda – $1e-6$) kiritadi;

- $[T, Y]= solver ('F', tspan, y_0, options, p_1, p_2, \dots)$ – ishlash prinsipi xuddi yuqoridagi formatlar kabi, faqat qo'shimcha p_1, p_2, \dots parametrlar F nomi bilan m -faylga uzatiladi (har bir murojaatda). Agar hech qanday parametrlar ishlatilmasa, $options$ o'rniga bo'sh matrisa “[]” belgisini qo'yish kerak;

- $[T, Y, TE, YE, IE]= solver ('F', tspan, y_0, options)$ – yechimga qo'shimcha ravishda events hossalari beradi. Ular odefile da aniqlangan hodisa funksiyalariga murojaat orqali $options$ strukturasi o'rnatilgan (Odefile shunday yozilish kerakki, u zarur informatsiyani qaytarsin). TE - hodisalar ro'y bergan momentlar vektor-ustuni; YE – TE larga mos keluvchi echim, IE vektordagi indekslar TE da aniqlangan funksiyalardan nolga tenglarini ko'rsatadi.

Agar funksiya chiqish parametrlarsiz ko'rsatilsa, u holda avtomatik ravishda hisoblangan echimni qurish uchun odeplot funksiyasi chaqiriladi.

- [T, X, Y]= solver ('model', tspan, y₀, options, ut, p1, p2, ...) - Simulink modelini ishlatadi.

21.4.Options parametrlarining qo'llanishi

Integrallash parametrlari ("options") m-fayllarda ham, odeset komandasi orqali buyruqlar oynasida ham aniqlanishi mumkin. Agar ikkala joyda ham aniqlangan bo'lsa, buyruqlar oynasidagi aniqlanish ustuvordir.

Har bir "yechgich" ma'lum bir parametrlarni ishlatadi. Quyida parametrlar tavsiflarini keltiramiz:

1. RelTol – musbat skalyar bo'lib, tanlashning nisbiy chegarasini bildiradi; barcha "yechgich" larda avtomatik tarzda $1e-3$ ga teng (0.1% aniqlik); iteratsiyaning har bir qadamida xatolik bahosi $e(i) \leq \max(\text{RelTol} * \text{abs}(y(i)), \text{AbsTol}(i))$;

2. AbsTol – absolyut aniqlik (musbat skalyar yoki vektor $\{1e-6\}$). Skalyar echim vektorining barcha komponentalari uchun kirgiziladi, vektor esa yechim komponentalari uchun ko'rsatiladi, barcha yechgichlar uchun AbsTol avtomatik tarzda $1e-6$ ga teng.

3. Norm Control – echim vektori normasiga boq'liq holda xatolikni boshqarish [on {off}] $\text{norm}(e) \leq \max(\text{RelTol} * \text{norm}(y), \text{AbsTol})$ bo'lishi uchun "on" o'rnatilishi kerak. Barcha yechgichlar avtomatik tarzda echim vektorining komponentalariga qattiqroq boshqarish qo'llaydi.

4. Refine – chiqish aniqligi faktori (musbat butun son) – chiqish nuqtalari sonini shu songa ko'paytiradi; avtomatik tarzda 1(faqat ODE 45 da 4 ga teng). Agar tspan>2 bo'lsa, qo'llash mumkin emas.

5. OutPutFsn – chiqish funksiyasi deskriptori; chiqish funksiyasi ko'rsatilmasdan "yechgich" ga murojaat etilganda ishlatiladi; avtomatik tarzda odeplot funksiyasini chaqiradi.

6. OutPutSel – tanlash indeksleri (butun sonlar vektori); OutPut Fsn ga kiruvchi komponentalarini o'rnatish kerak; avtomatik tarzda barcha komponentalarni chiqaradi.

7. Stats – [on {off}] hisoblashlar qiymatlari statistikasini ko'rsatish kerak;

8. Jacobion – Yakobi matritsasi funksiyasi [function / constant matrix]; FJac funksiya deskriptoriga (agar FJac dF/dy ni qaytarsa) yoki o'zgarmas dF/dy matritsa nomiga o'rnatish kerak;

9. Jpattern - Yakobi matritsasining siyraklashtirilganlik grafigi (siyraklashgan matritsa nomi).

10. Vectorised – vektorlashtirilgan ODE funksiya [on {off}]; agar $FF(t, [y_1, y_2, \dots])$ ODE – funksiya $[F(t, y_1), F(t, y_2) \dots]$ vektorini qaytaradigan bo'lsa, “on” rejimi o'rnatiladi.

11. Events – [function] – hodisalar funksiyalarining deskriptorlari kiritiladi.

12. Mass – massalar matritsasi [constant matrix / function] $M * y' = f(t, y)$ tenglama uchun o'zgaruvchan M uchun massalar matritsasi funksiyasining deskriptori kiritiladi.

13. M state Dependence – massalar matritsasining u ga bog'liqligi [none {weak / strong}]. $M * y' = f(t, y)$ tenglama uchun none ni o'rnatish kerak. “Kuchsiz” (<weak>) hamda “kuchli” () boq'liqliklar M (t, y) ni anglatadi; “weak” holda yechimning oshkormas algoritmlari qo'llaniladi.

14. Mass Singular – M massalar matritsasi singulyar. [yes |no| {maybe} |da|нет|может быть];

15. MV Pattern – siyraklashganlik (dMV/dy), siyraklashganlik grafigi (Spy funksiyasiga qarang) – ixtiyoriy R uchun $ScS(i, j) = 1$ siyraklashgan matritsa nomi kiritiladi, bu erda M(t, y) matritsaning (i, k) elementi y o'zgaruvchining j proeksiyasiga boq'liq, aks holda 0.

16. Initialslope – boshlang'ich oq'ish vektorini $ur_0 = F(t_0, y_0) / M(t_0, y_0)$;

17. Initialstep – qadamning boshlang'ich o'lchami, avtomatik tarzda “yechgich” o'z algoritmi bo'yicha belgilashi mumkin.

18. MaxStep – maksimal qadam, avtomatik tarzda barcha “yechgich” larda tspan/10 ga teng.

19. BDF (Backward Differentiation Formulas) [on {off}] – ode 15s da avtomatik tarzda qo'llaniladigan sonli differensiallash formulalari o'rniga teskari differensiallash formulalarini (Gear metodlari) qo'llash kerakligini ko'rsatadi.

20. Max Order – ode 15s ning maksimal tartibi [1|2|3|4|{5}].

Quyida parametrlarning “yechgich” larda qo'llanish jadvali keltirilgan:

№	Parametrlar	ode 45	ode 23	ode 113	ode 159	ode 239
1.	Rel Tol, Abs Tol	+	+	+	+	+
2.	Output Fcn, Output	+	+	+	+	+

	Sel, Refine, Stats					
3.	Events	+	+	+	+	+
4.	Max Step, Intial Step	+	+	+	+	+
5.	Y constant, Jacobion, Jpattern, Vectorised	-	-	-	+	+
6.	Mass	-	-	-	+	+
7.	Mass Sonstant	-	-	-	+	-
8.	Max Order, BDF	-	-	-	+	-

Parametrlarni kiritish uchun odeset funksiyasidan foydalanish zarur:
Options = odeset ('name 1', 'value 1', 'name 2', 'value 2', ...)

Bu komanda yordamida integrallovchining parametrlari strukturasi yaratiladi (ko'rsatilgan parametrlar ko'rsatilgan qiymatlarni oladi). Barcha aniqlanmagan parametrlar bo'sh matrisa [] qiymatini oladi. YEchgich bvp4c kam sonli parametrlarga ega bo'lsa-da, integrallanuvchi funksiyaning Yakobi matritsasidan tashqari yana chegaraviy shartlardagi va noma'lum parametrlar bo'yicha funksiyalarning xususiy hosilalarini ham o'z ichiga oluvchi Yakobi matrisatsini kiritish mumkin.

"Yechgich" lar echimlarning oddiy grafiklarini ham, fazali portretlarini (yani bir o'qda biror boq'liqlik, boshqa o'qda uning hosilasi ko'rsatilgan parametrik grafiklar) ham qurish imkoniyatini beradi. Masalan, tebranishning statsionar sinusoidasining fazali portreti ellips yoki aylanadir.

21.5. Differensial tenglamalarni yechishga doir misollar

1-misol. Yuqoriga otilgan qattiq jismning erkin tushish (xavo qarshiligini hisobga olmagan holda) harakatini ifodalovchi

$$y''(t) = -g$$

tenglamani qaraylik. Bu tenglamani $y_1'=y_2$, $y_2'=y_1'$ belgilashlar orqali quyidagi

$$\begin{cases} y_1' = y_2 \\ y_2' = -g \end{cases} \quad (1)$$

tenglamalar sistemasiga (ODTS) olib kelish mumkin (bu yerda y_1 – balandlik, y_2 – tezlik, $g=9.8$ m/sek erkin tushish tezligi). (1) tenglamalar sistemasining ushbu

$$\begin{cases} y_1(0) = 0 \\ y_2(0) = 10 \end{cases}$$

boshlang'ich shartlarni qanoatlantiruvchi echimini topish talab etiladi.

Yechish. Demak, $y_0 = [0; 10]$ – boshlanq'ich shartlar vektor-ustuni.

Endi démodé nomli script-fayl tuzamiz va saqlab qo'yamiz:

```
y0 = [0; 10] % boshlang'ich shartlar
```

```
ts = 0:.2:2 % vaqt intervali
```

```
dydt = @(t,y) [y(2); -9.8] % ode ong qismning anonim funksiyasi
```

```
[t0, y0] = ode45(dydt, ts, y0) % ode45 yechgich
```

```
plot(t0, y0)
```

```
Command Window
```

```
>> demoode
```

```
y0 =
```

```
    0
```

```
   10
```

```
ts =
```

```
Columns 1 through 5
```

```
    0    0.2000    0.4000    0.6000    0.8000
```

```
Columns 6 through 10
```

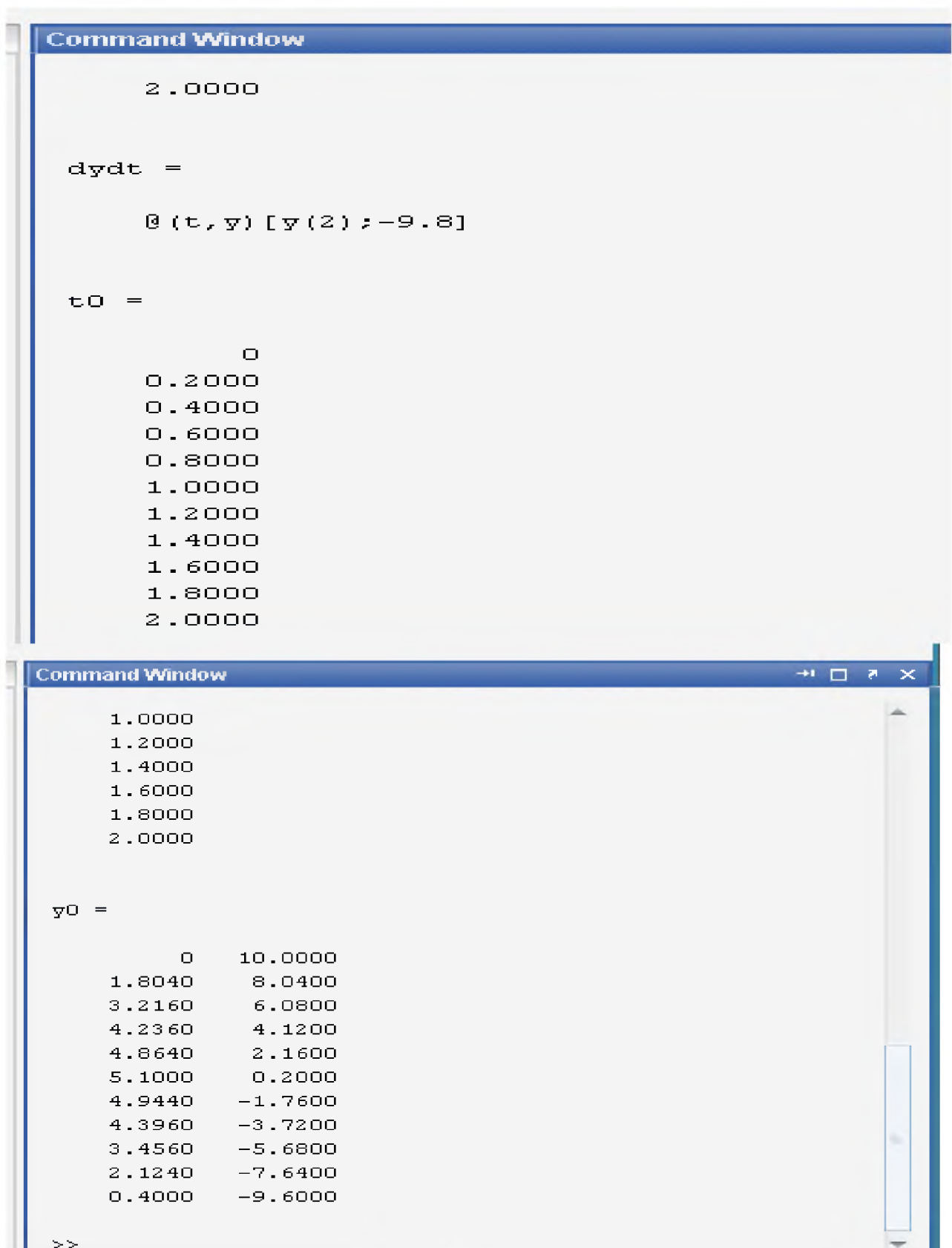
```
    1.0000    1.2000    1.4000    1.6000    1.8000
```

```
Column 11
```

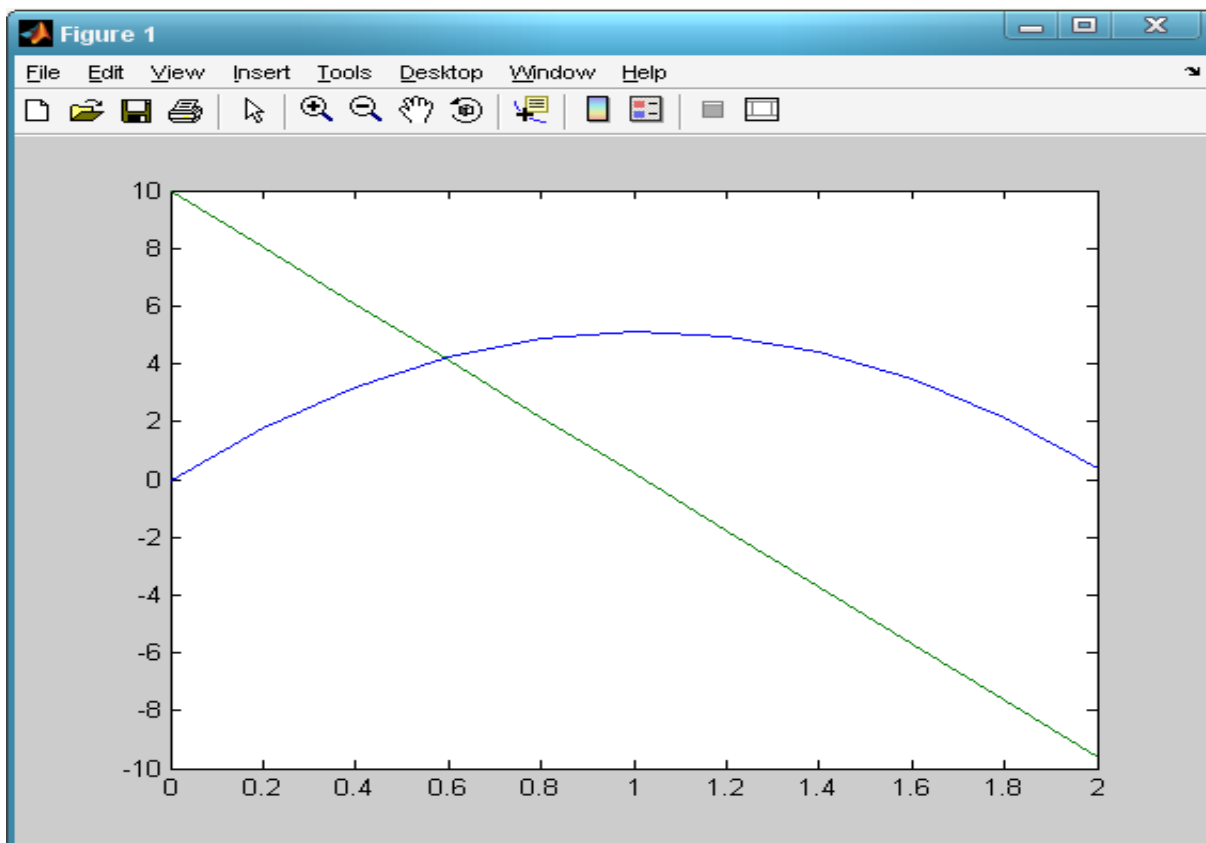
```
    2.0000
```

OVR

21.1 - rasm. Démodé faylidan olingan natija.



21.2- rasm. Démodé faylidan olingan natija.



21.3- rasm. Taqribiy yechimning grafigi.

2-misol. Differensial tenglamalar sistemasi (2-tartibli noxiziqli differensial tenglama - Van-der- Pol tenglamasi)

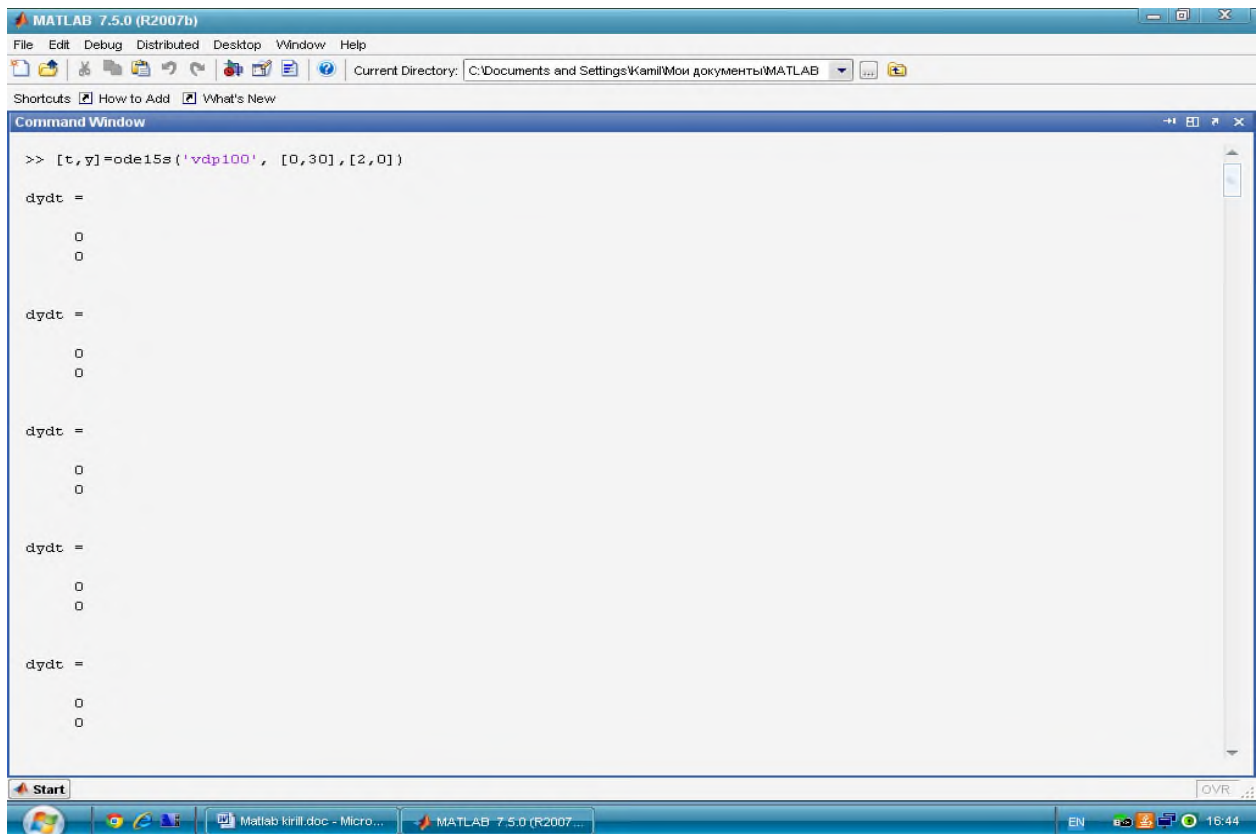
$$\begin{cases} y_1' = y_2; \\ y_2' = m \cdot (1 - y_1^2) \cdot y_2 - y_1 \end{cases}$$

Ning quyidagi $y(0) = 0$; $y_2(0) = 1$ boshlang'ich shartlar asosida yechimni toping.

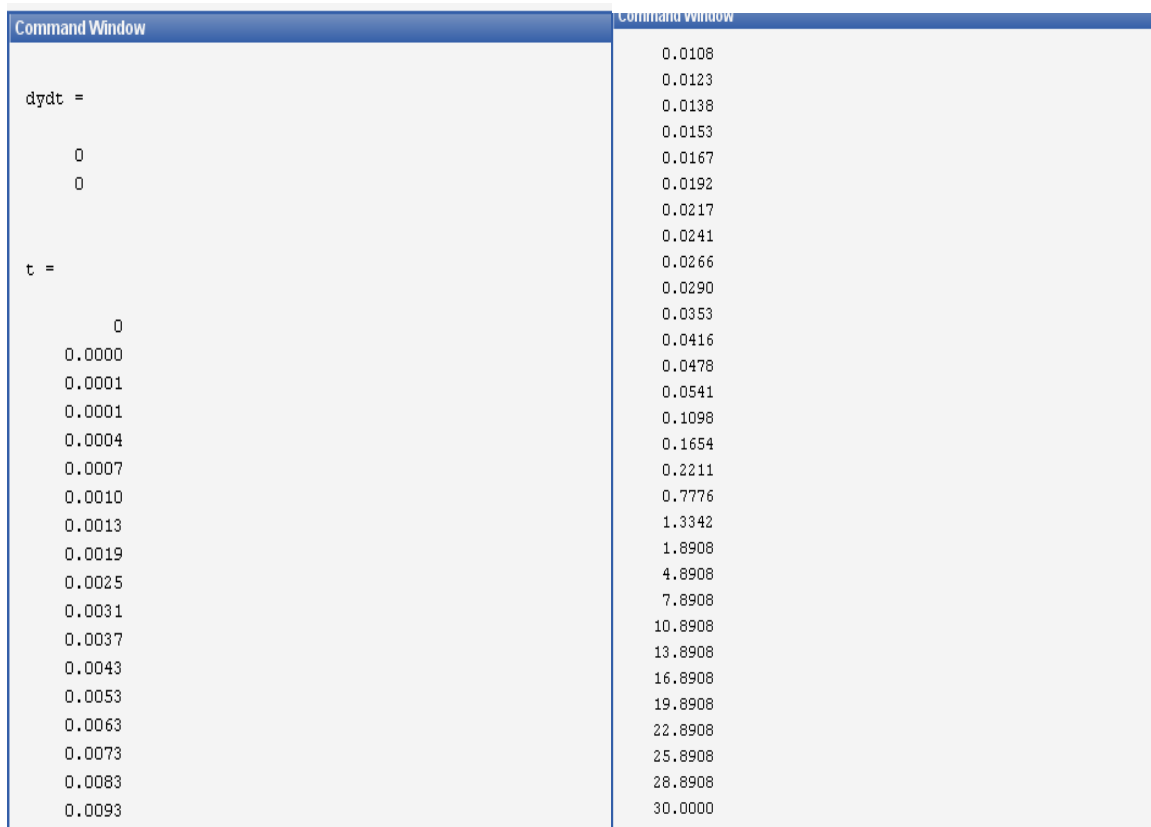
Yechish. Sistema holati m -parametr qiymatiga boq'liq. Agar m katta qiymat qabul qilsa, sistema qattiq bo'ladi. Biz $m = 100$ deb olamiz.

Avval sistemani ODE funksiya ko'rinishda yozib olish kerak. Buning uchun asosiy menyuda File=>New=>M-file tanlab quyidagilarni kiritamiz (yani vdp100 nomli fayl- funksiya yaratamiz va saqlaymiz):

```
function dydt=vdp100(t,y)
dydt=zeros(2,1)% Vector – ustun
dydt(1)=y(2);dydt(2)=100*(1-y(1)^2)*y(2)-y(1);
Endi ode 15s “yechgich”ni qo'llaymiz:
>>[t,y]=ode 15s('vdp100', [0,30],[2,0])
```



21.4 - rasm. MATLAB yechgichining qo'llanishi.



21.5 - rasm. MATLAB yechgichining qo'llanishi.

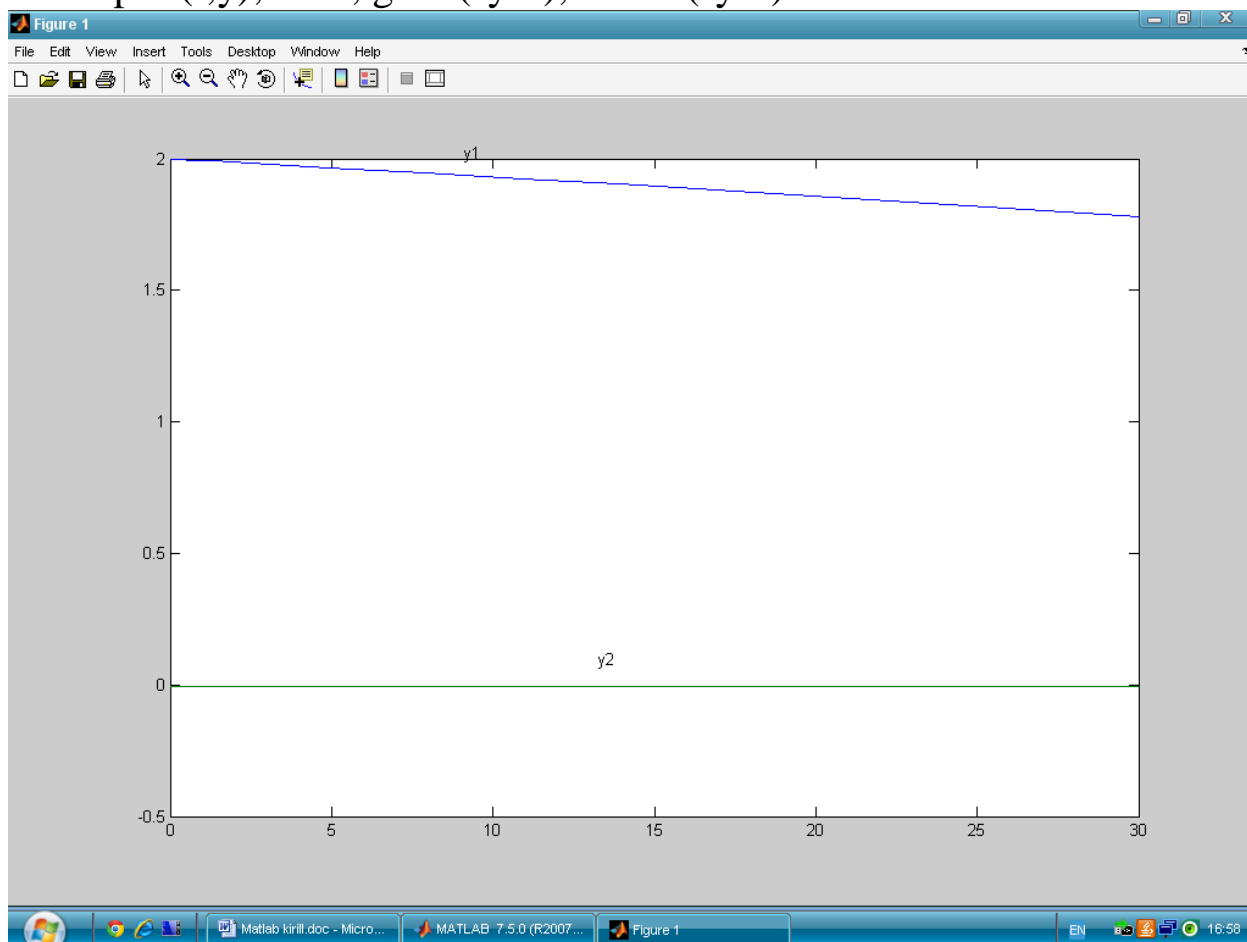
```
Command Window
25.8908
28.8908
30.0000

y =
2.0000    0
2.0000  -0.0001
2.0000  -0.0002
2.0000  -0.0003
2.0000  -0.0008
2.0000  -0.0013
2.0000  -0.0017
2.0000  -0.0021
2.0000  -0.0029
2.0000  -0.0035
2.0000  -0.0040
2.0000  -0.0045
2.0000  -0.0048
2.0000  -0.0053
2.0000  -0.0057
2.0000  -0.0059
2.0000  -0.0061
2.0000  -0.0063
1.9999  -0.0064
1.9999  -0.0065
1.9999  -0.0066
1.9999  -0.0066
1.9999  -0.0066
1.9999  -0.0066
1.9999  -0.0067
2.0000  -0.0063
1.9999  -0.0064
1.9999  -0.0065
1.9999  -0.0066
1.9999  -0.0066
1.9999  -0.0066
1.9999  -0.0066
1.9999  -0.0067
1.9999  -0.0067
1.9998  -0.0067
1.9998  -0.0067
1.9998  -0.0067
1.9997  -0.0067
1.9997  -0.0067
1.9997  -0.0067
1.9993  -0.0067
1.9989  -0.0067
1.9985  -0.0067
1.9948  -0.0067
1.9911  -0.0067
1.9873  -0.0067
1.9669  -0.0069
1.9462  -0.0070
1.9250  -0.0071
1.9035  -0.0073
1.8815  -0.0074
1.8590  -0.0076
1.8361  -0.0077
1.8125  -0.0079
1.7885  -0.0081
1.7794  -0.0082
```

21.6 - rasm. MATLAB yechgichining qo'llanishi.

Agar yechim grafiklarini ko'rish kerak bo'lsa, zarur komanda beriladi:

```
>>plot(t,y); hold; gtext('y1'), text('y2')
```



21.7 - rasm. Yechim grafiklari.

Bu yerda gtext komandasi “sichqoncha” yordamida grafiklarga “y1” va “y2” yozuvlarini qo’yish imkonini beradi.

Nazorat savollari

1. Differensial tenglama deb qanday tenglamaga aytiladi?
2. Differensial tenglama tartibi deganda nimani tushunasiz?
3. ODT ning yechim tushunchasini ayting.
4. Xususiy yechim tushunchasini keltiring.
5. Boshlanq’ich shartlar qanday qo’yiladi?
6. ODTS ni echish uchun qanday MATLAB yechgichlari bor?

Mustaqil ishlash uchun misollar

- 1) $xy' - y = 0,$ $y(-2) = 4$
- 2) $yy' + x = 0,$ $y(1) = 5$
- 3) $x^2 y' + y^2 = 0,$ $y(-1) = 1$

- 4) $y' = (2y + 1)\operatorname{ctgx}$, $y(1) = 3$
5) $(1 + x^2)y' + y = xy$, $y(0) = 1$
6) $x^3 y'' + x^2 y' = 1$, $y(0) = -1, y'(-1) = 1$
10) $y''' = 4\cos 2x$, $y(0) = 0, y'(0) = 0$

ADABIYOTLAR

1. Хорн Р., Джонсон Ч. Матричный анализ. — М.: Мир, 1989.
2. Дьяконов В., Абраменкова И. MATLAB. Обработка сигналов и изображений. Специальный справочник. СПб.: Питер, 2002.
3. Потемкин В.Г. Вычисления в среде MATLAB / В.Г. Потемкин. М.: ДИАЛОГ-МИФИ, 2004.
4. Кетков Ю.Л. MATLAB 7: Программирование, численные методы / Ю.Л. Кетков, А. Ю. Кетков, М. М. Шульц. - СПб: БХВ-Петербург, 2005.
5. Гультияев. А. Визуальное моделирование в среде MATLAB: Учебный курс. – СПб.: Питер, 2000.
6. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6. -. М.: ДИАЛОГ-МИФИ, 2002.-496 с.
7. MATLAB. The language of Technical Somputing. Getting Started with MATLAB. The Math Works, Inc. USA, 2000.
8. MATLAB. The Language of Technical Somputing. Using MATLAB Graphics. The Math Works, Inc. USA, 2000.
9. Ильин В.А., Позняк Э.Г. Основы математического анализа. Учеб. для вузов в 2-х частях. – 6-е изд. стер. –М. Физматлит, 2002, -646 с.
10. Марчук Г.И. Методы вычислительной математики. М.: Наука, 1989, 608 с.
11. Бахвалов Н.С., Жидков Н. П., Кобельков Г. М. Численные методы. М.: Наука. Физматлит. — 1987.
12. Дьяконов В. П. MATLAB. Полный самоучитель. – М.: ДМК Пресс, 2012. – 768 с.
13. Мещеряков, В.В. Задачи по математике с MATLAB & Simulink / В.В. Мещеряков. - Москва : Диалог-МИФИ, 2007. - 528 с.
14. Воеводин В.В. Линейная алгебра. М.: Новосибирск: Наука, 1980.
15. Dadajonov T., Muhitdinov M. MATLAB asoslari.- T.:- Fan, 2008

QO'SHIMCHA ADABIYOTLAR

1. Мирзиёев Ш.М. Буюк келажакимизни мард ва олижаноб халқимиз билан бирга курамиз. 2017.
2. Мирзиёев Ш.М. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. 2017.

3. Мирзиёев Ш.М. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. 2017.
4. Мирзиёев Ш.М. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қоидаси бўлиши керак. Ўзбекистон Республикаси Вазирлар Маҳкамасининг 2016 йил якунлари ва 2017 йил истиқболларига бағишланган мажлисидаги Ўзбекистон Республикаси Президентининг нутқи. // Халқ сўзи газетаси. 2017 йил 16 январ, № 11.

MUNDARIJA

KIRISH	3
1. MALAB ishchi stoli va asosiy obyektlari	8
1.1. Matlab ishchi stoli.....	8
1.2. Tizim kengaytmasi. Yordam tizimi	10
1.3. Matlabning asosiy ob'ektlari, funksiyalari va sozlangan fuyrsiyalari.....	12
Nazorat savollari	16
2. MA'LUMOTLARNI KIRITISH VA ODDIY HISOBLASH QOIDALARI.....	17
2.1. Ma'lumotlarni(matritsalar)ni kiritish.....	17
2.2. Ma'lumotlar(matritsalar) va ularni shakllantirish usullari	18
3. MATLABDA VEKTORLAR VA MATRITSALAR USTIDA AMALLAR	23
3.1. Skalyar miqdorlar ustida arifmetik amallar	23
3.2. Matritsalar ustida oddiy arifmetik amallar	23
3.3. Matlabda massivlar ustida maxsus amallar	24
3.4. Vektorlar ustida maxsus amallar	24
3.5. Mantiqiy amallar.....	25
3.6. Matritsalarini almashtirish amallari	27
3.7. Sana va vaqt funksiyalari.....	31
Nazorat savollari	32
4. MATLABDA SIYRAKLASHGAN MATRITSALAR.....	34
4.1. Siyraklashgan matritsalar ustida amallar bajarish	34
Nazorat savollari.....	41
5. SIMVOLLI O'ZGARUVCHILAR ALGEBRASI.....	42
5.1. Simvolli funksiyalar va ifodalar	42
5.2. Simvolli o'zgaruvchilar yordamida algebraik tenglamalarni yechish.....	46
Nazorat savollari	52
Mustaqil ishlash uchun misollar	53

6. KO'PHADLAR BILAN ISHLASH	54
6.1. Ko'phadlar bilan boq'liq amallar	54
6.2. Ko'phadlarning qiymatlarini va ildizlarini hisoblash.....	55
6.3 . Xarakteristik ko'phadlar	57
6.4. Ko'phadlarni ko'paytirish va bo'lish.....	58
6.5. Ko'phadlarning hosilasini hisoblash	59
Nazorat savollari	61
Mustaqil ishlash uchun misollar	61
7. DASTURLASH ASOSLARI. MATLABDA MA'LUMOTLAR VA FAYLLARNING TOIFA(TIP)LARI.....	62
7.1. Matlabda dasturlash vositalari	62
7.2. Matlabda ma'lumotlar toifalari.....	62
7.3. Fayllarning toifalari	64
7.4. Ssenariy fayllari (Script-fayl) tuzilishi va xossalari	66
7.5. Fayl-funksiya va uning xossalari	68
7.6. Lokal va global o'zgaruvchilar	72
7.7. O'zgaruvchi sondagi argumentli funksiyalar	74
Nazorat savollari	78
Mustaqil ishlash uchun misollar	78
8. MATLABDA DASTURLASH ASOSLARI. SHARTLI VA SIKL OPERATORLARI	80
8.1. Sikl operatorlari	80
8.2. Tayinlash va shartli operatorlar	83
8.3. Tanlash operatori	85
8.4. Hisoblashlarda pauzalar hosil qilish.....	87
Nazorat savollari	88
9. DASTURNI SOZLASH	89
9.1. Dasturni sozlash komandalari.....	89
9.2. m-fayl listingi satrlarini raqamlab chiqarish	89
9.3. Uzilish nuqtalarini o'rnatish , olib tashlash va ko'rib chiqish .	90
9.4. m-faylni bajarilishini boshqarish.....	91

9.5. Ishchi fazoni ko'rish	91
9.6. m-fayllarni profillash	92
Nazorat savollari	93
10. XATOLIKLARNI QAYTA ISHLASH	94
10.1. Xatoliklar haqidagi axborot	94
10.2. Xatoliklarni bildiruvchi error va warning komandalari	96
10.3. Lasterr funksiyasi va xatoliklarni qayta ishlash	96
10.4 . varargin va varargout o'zgaruvchilari	97
10.5. M-fayl funksiyalarni bajarilish xususiyatlari va izohlar haqida	98
10.6. P-kodlarni yaratish.....	100
Nazorat savollari	101
11. OB`YEKTGA MO`LJALLANGAN DASTURLASH	102
11.1. Ob`yektning sinfini tekshirish	102
11.1. Obyektning sinfini tekshirish.....	103
11.2. Handle va inline funksiyalar	104
Nazorat savollari	107
Mustaqil ishlash uchun misollar	107
12. MATLABDA GRAFIK VA GISTOGRAMMALAR	108
12.1. Matlabda oddiy grafik.....	108
12.2. Gistogrammalar	112
12.3. Polyar koordinatalarda grafik	114
12.4. Uch o'lchovli grafika	116
12.5. Bir nechta grafiklarni hosil qilish	119
12.6. Silindr va sferani qurish.....	120
Nazorat savollari	126
Mustaqil ishlash uchun misollar	127
13. MAXSUS GRAFIKA. ANIMATSIYA BAJARISH VOSITALARI.....	128
13.1. Animatsiyani bajarish vositalari	128
13.2. Nuqtaning fazoda harakatlanishi	129

13.3. Deskriptorli grafika.....	130
13.4. Ob'yektlar deskriptorlari	132
13.5. Foydalanuvchi interfeysini yaratish	133
13.6. Uch o'lchovli grafiklar galeriyasi.....	137
Nazorat savollari	138
Mustaqil ishlash uchun misollar	138
14. MATLAB PAKETINING KENGAYTMASI, BIBLIOTEKALAR.....	139
14.1. MATLAB strukturasi.....	139
14.2. Image Processing bibliotekasi	141
14.3. Signal Processing bibliotekasi	141
14.4. Simulink va Stateflow paketi.....	142
Nazorat savollari	143
15. SIMULINK PAKETI–DINAMIK TIZIMLARNI VIZUAL MODELLASHTIRISH TIZIMI.....	145
15.1. Simulink paketining umumiy vazifalari	145
15.2. Modellashtirishda Simulink paketining roli	145
15.3.Stateflow programmasi.....	146
Nazorat savollari	147
16. CHIZIQLI TENGLAMALAR SISTEMASINI MATLAB MUHITIDA YECHISH	148
16.1. Chizikli tenglamalar sistemasini.....	148
16.2. Chizikli tenglamalar sistemasini yechish usullari	149
16.3. Chizikli tenglamalar sistemasini echishda MATLAB usullari	150
16.4. Chizikli tenglamalar sistemasini yechishga doir misollar.....	151
Nazorat savollari	156
Mustaqil ishlash uchun misollar	157
17. KUZATISH (STATISTICA) NATIJALARINI BIRLAMCHI QAYTA ISHLASH	158
17.1. Ma'lumotlarni statistik qayta ishlash masalasi.....	158
17.2. Strukturali identifikatsiya	158

17.3. Parametrik identifikatsiya	160
17.4. Ma'lumotlarni statistik qayta ishlash uchun Matlabning asosiy funksiyalari	162
Nazorat savollari	167
18. BIR VA KO'P O'ZGARUVCHILI FUNKSIYALAR UCHUN OPTIMALLASHTIRISH.....	168
18.1. Funksiyalar uchun optimallashtirish masalasining qo'yilishi	168
18.2. Funksiyalar uchun optimallashtirish masalasini yechish usullari	168
18.3. Optimallashtirish masalasini echish uchun MATLAB funksiyalari	169
18.4. Funksiya ekstremumini topishga doir misollar	172
18.5. Rozenbrok test funksiyasini minimallashtirish	177
18.6. Ko'p o'zgaruvchili funksiyani minimallashtirishning boshqa usullari	178
18.7. Optimizatsion kutubxonaning imkoniyatlari.....	181
Mustaqil ishlash uchun misollar	181
Nazorat savollari	182
19. FUNKSIYA HOSILASINI CHEKLI AYIRMALAR BILAN APPROKSIMATSIYALASH VA SONLI INTEGRALLASH MASALALARI.....	183
19.1 Chekli ayirmalar	183
19.2. Funksiya hosilasi	185
19.3. Funksiya gradientini hisoblash	187
19.4. Sonli integrallash	189
19.5. Kvadraturalar usulida integrallash.....	190
Nazorat savollari	191
20. MATLAB FUNKSIYALARINI SIGNALLARNI RAQAMLI QAYTA ISHLASH MASALALARIGA QO'LLANILISHI	193
20.1. Signallarni raqamli qayta ishlash tushunchasi (SRQI).....	193
20.2. Signallarni tahlil qilish va filtrlash	195
20.3. SRQI masalalarini echish uchun MATLAB muhiti	196

20.4. SRQI standart masalalarini yechishga doir misol	197
Nazorat savollari	201
21. MATLAB YORDAMIDA DIFFERENSIAL TENGLAMALARNI YECHISH.....	202
21.1. Differensial tenglamalarning matematik tavsifi	202
21.2. Oddiy differensial tenglamalar sistemasi(ODTS)ni yechish uchun MATLAB “Yechgich” lari.....	203
21.3. Differensial tenglamalarni yechish uchun funksiyalar.....	204
21.4.Options parametrlarining qo'llanishi.....	206
21.5. Differensial tenglamalarni yechishga doir misollar	208
Nazorat savollari	214
Mustaqil ishlash uchun misollar	214
ADABIYOTLAR.....	216

A.H.NISHANOV, A.T.RAHMANOV,
M.X.AKBAROVA.

AMALIY DASTURIY PAKETLAR

(O'quv qo'llanma)

Toshkent – «Aloqachi» – 2019

Muharrir: M.Mirkomilov
Tex. muharrir: A.Tog'ayev
Musavvir: B.Esanov
Musahhiha: F.Tog'ayeva
Kompyuterda
sahifalovchi: Sh.To'xtamurodov

Nashr.lits. AI №176. 11.06.11.
Bosishga ruxsat etildi: 3.06.2019. Bichimi 60x841 /16.
Shartli bosma tabog'i 14,5. Nashr bosma tabog'i 14,0.
Adadi 100. Buyurtma № 89.

«Nihol print» Ok da chop etildi.
Toshkent sh., M. Ashrafiy ko'chasi, 99/101.