

**O‘ZBEKISTON RESPUBLIKASI
OLIY TA’LIM, FAN VA INNOVATSIYALAR
VAZIRLIGI**

**BUXORO DAVLAT UNIVERSITETI HUZURIDAGI PEDAGOG
KADRLARNI QAYTA TAYYORLASH VA ULARNING
MALAKASINI OSHIRISH MINTAQAVIY MARKAZI**

“TASDIQLAYMAN”

Mintaqaviy markaz direktori

B.R. Adizov

“ ” 2025-yil

**“FIZIK JARAYONLARNI KOMPYUTERDA
MODELLASHTIRISH”**

moduli bo‘yicha

O‘QUV-USLUBIY MAJMUA

**Qayta tayyorlash va malaka
oshirish kursi yo‘nalishi:**

“Fizika va astronomiya”

Tinglovchilar kontingenti:
pedagoglari

Oliy ta’lim muassasalarining

Buxoro – 2025

Modulning o`quv-uslubiy majmuasi Oliy ta'lim, fan va innovatsiyalar vazirligining 2024 yil 27 dekabrda 485-sonli buyrug'i bilan tasdiqlangan o`quv dasturi va o`quv rejasiga muvofiq ishlab chiqilgan.

Tuzuvchi: Sh.E.Nazarov – PhD, dotsent.

Taqrizchi: T.R.Shafiyev – PhD, dotsent

**O`quv -uslubiy majmua Buxoro davlat universiteti Ilmiy
Kengashining qarori bilan nashrga tavsiya qilingan
(2024 yil “27” dekabrda 5-sonli bayonnoma)**

MUNDARIJA

ISHCHI DASTUR.....	4
MODULNI O'QITISHDA FOYDALANILADIGAN	
INTERFAOL TA'LIM METODLARI.....	14
NAZARIY MASHG'ULOTLAR.....	18
AMALIY MASHG'ULOTLAR.....	90
GLASSARIY.....	112
ADABIYOTLAR RO'YXATI.....	121

I. ISHCHI DASTUR

Kirish

Mazkur ishchi o'quv dasturi O'zbekiston Respublikasining 2020-yil 23-sentabrda tasdiqlangan "Ta'lim to'g'risida" Qonuni, O'zbekiston Respublikasi Prezidentining 2015-yil 12-iyundagi "Oliy ta'lim muassasalarining rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini yanada takomillashtirish to'g'risida" PF-4732-son, 2019-yil 27-avgustdagi "Oliy ta'lim muassasalari rahbar va pedagog kadrlarining uzluksiz malakasini oshirish tizimini joriy etish to'g'risida" PF-5789-son, 2019-yil 8-oktabrdagi "O'zbekiston Respublikasi oliy ta'lim tizimini 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to'g'risida" PF-5847-son, 2020-yil 29-oktabrdagi "Ilm-fanni 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to'g'risida" PF-6097-son, 2022-yil 28-yanvardagi "2022-2026 yillarga mo'ljallangan Yangi O'zbekistonning taraqqiyot strategiyasi to'g'risida" PF-60-son, 2023-yil 25-yanvardagi "Respublika ijro etuvchi hokimiyat organlari faoliyatini samarali yo'lga qo'yishga doir birinchi navbatdagi tashkiliy chora-tadbirlar to'g'risida" PF-14-son, O'zbekiston Respublikasi Prezidentining 2023-yil 11-sentabrdagi "O'zbekiston - 2030" strategiyasi to'g'risida" PF-158-son Farmonlari, shuningdek, O'zbekiston Respublikasi Prezidentining 2024-yil 21-iyundagi "Aholi va davlat xizmatchilarining korrupsiyaga qarshi kurashish sohasidagi bilimlarini uzluksiz oshirish tizimini joriy qilish chora-tadbirlari to'g'risida" PQ-228-son, O'zbekiston Respublikasi Prezidentining 2021-yil 17-fevraldagi "Sun'iy intellekt texnologiyalarini jadal joriy etish uchun shart-sharoitlar yaratish chora-tadbirlari to'g'risida" PQ-4996-son qarorlari va O'zbekiston Respublikasi Vazirlar Mahkamasining "Oliy ta'lim muassasalari rahbar va pedagog kadrlarining malakasini oshirish tizimini yanada takomillashtirish bo'yicha qo'shimcha chora-tadbirlar to'g'risida" 2019-yil 23-sentabrdagi 797-son hamda O'zbekiston Respublikasi Vazirlar Mahkamasining "Oliy ta'lim tashkilotlari rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini samarali tashkil qilish chora-tadbirlari to'g'risida" 2024-yil 11-iyuldagi 415-son Qarorlarida belgilangan ustuvor vazifalar mazmunidan kelib chiqqan holda tuzilgan bo'lib, u oliy ta'lim muassasalari pedagog kadrlarining kasb mahorati hamda innovatsion kompetentligini rivojlantirish, sohaga oid ilg'or xorijiy tajribalar, yangi bilim va malakalarni o'zlashtirish, shuningdek amaliyotga joriy etish ko'nikmalarini takomillashtirishni hamda rivojlangan xorijiy davlatlarning oliy ta'lim sohasida erishgan yutuqlari hamda orttirgan tajribalari asosida "Fizika va astronomiya" qayta tayyorlash va malaka oshirish yo'nalishi uchun tayyorlangan namunaviy o'quv reja hamda dastur mazmunidan kelib chiqqan holda tuzilgan bo'lib, u zamonaviy talablar asosida qayta tayyorlash va malaka oshirish jarayonlarining mazmunini takomillashtirish hamda oliy ta'lim muassasalari pedagog kadrlarining kasbiy kompetentligini muntazam oshirib borishni maqsad qiladi.

Jamiyat taraqqiyoti nafaqat mamlakat iqtisodiy salohiyatining yuksakligi bilan, balki bu salohiyat har bir insonning kamol topishi va uyg'un rivojlanishiga qanchalik

yoʻnaltirilganligi, innovatsiyalarni tadbiq etilganligi bilan ham oʻlchanadi. Demak, taʼlim tizimi samaradorligini oshirish, pedagoglarni zamonaviy bilim hamda amaliy koʻnikma va malakalar bilan qurollantirish, chet el ilgʻor tajribalarini oʻrganish va taʼlim amaliyotiga tadbiq etish bugungi kunning dolzarb vazifasidir. “Fizik jarayonlarni kompyuterda modellashtirish” moduli aynan mana shu yoʻnalishdagi masalalarni hal etishga qaratilgan.

Modulning maqsadi va vazifalari

Oʻquv modulning maqsadi: pedagog kadrlarning innovatsion yondoshuvlar asosida oʻquv-tarbiyaviy jarayonlarni yuksak ilmiy-metodik darajada loyihalashtirish, sohadagi ilgʻor tajribalar, zamonaviy bilim va malakalarni oʻzlashtirish va amaliyotga joriy etishlari uchun zarur boʻladigan kasbiy bilim, koʻnikma va malakalarini takomillashtirish, shuningdek ularning ijodiy faolligini rivojlantirishdan iborat

Oʻquv modulning vazifalari:

- “Fizika va astronomiya” yoʻnalishida pedagog kadrlarning kasbiy bilim, koʻnikma, malakalarini takomillashtirish va rivojlantirish;
- pedagoglarning ijodiy-innovatsion faollik darajasini oshirish;
- pedagog kadrlar tomonidan zamonaviy axborot-kommunikatsiya texnologiyalari, zamonaviy taʼlim va innovatsion texnologiyalar sohasidagi ilgʻor xorijiy tajribalarning oʻzlashtirilishini taʼminlash;
- oʻquv jarayonini tashkil etish va uning sifatini taʼminlash borasidagi ilgʻor xorijiy tajribalar, zamonaviy yondashuvlarni oʻzlashtirish;

“Fizika va astronomiya” yoʻnalishida qayta tayyorlash va malaka oshirish jarayonlarini fan va ishlab chiqarishdagi innovatsiyalar bilan oʻzaro integratsiyasini taʼminlash.

Modul boʻyicha tinglovchilarning bilimi, koʻnikmasi, malakasi va kompetensiyalariga qoʻyiladigan talablar

“Fizik jarayonlarni kompyuterda modellashtirish” modulini oʻzlashtirishda amalga oshiriladigan masalalar doirasida:

Tinglovchi:

- ✓ kompyuterli eksperiment bosqichlarini;
- ✓ dasturiy taʼminot: Spyder/MatlabWork vositalari va tillarini;
- ✓ kodlarni tuzish: algoritmlar, oʻzgaruvchanlar, maʼlumotlar turlarini;
- ✓ dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollarini;
- ✓ tarmoqlanuvchi algoritmlar, if va else shart operatrlari bilan ishlash misollarini **bilishi** kerak.

Tinglovchi:

- kompyuterda, modellashtirish uchun operatsion tizim tushunchasini bilish;
- Python redaktorlarini (Spyder, PyCharm, VS Code) oʻrnatish;
- munosabatlar, tanlash va solishtirish operatorlari bilan ishlash;

- sonli integrallash va differensial tenglamalarni sonli yechish usullardan foydalanish;
- teylor qatorlari asosida funksiyalarning qiymatini hisoblash;
- grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish **ko'nikmalariga** ega bo'lishi lozim.

Tinglovchi:

- ✓ sikllar yordamida summa va ko'paytmalarni topish;
- ✓ chala tarmoqli va to'liq tarmoqli algoritmlarni ahamiyatini ochib berish;
- ✓ python tiliga kirish: proyektni saqlash va tayyorini davom ettirish, sintaksisining asosiy qoidalarini qo'llash **malakalariga** ega bo'lishi lozim.

Tinglovchi:

- mexanik harakatlarni modellashtirish va ularni visuallashtirish usullaridan foydalanish **kompetensiyalariga** ega bo'lishi lozim.

Modulni tashkil etish va o'tkazish bo'yicha tavsiyalar

“Fizik jarayonlarni kompyuterda modellashtirish” moduli ma'ruza va amaliy mashg'ulotlar shaklida olib boriladi. Modulni o'qitish jarayonida ta'limning zamonaviy metodlari, pedagogik texnologiyalar va axborot-kommunikatsiya texnologiyalari qo'llanilishi nazarda tutilgan:

- ma'ruza darslarida zamonaviy kompyuter texnologiyalari yordamida prezentatsion va elektron-didaktik texnologiyalardan;
- o'tkaziladigan amaliy mashg'ulotlarda texnik vositalardan, ekspress-so'rovlar, test so'rovlari, aqliy hujum, guruhli fikrlash, kichik guruhlar bilan ishlash, kollokvium o'tkazish, va boshqa interaktiv ta'lim usullarini qo'llash nazarda tutiladi.

Modulning o'quv rejadagi boshqa modullar bilan bog'liqligi va uzviyligi

“Fizik jarayonlarni kompyuterda modellashtirish” moduli mazmuni o'quv rejadagi “Pedagogik faoliyatda raqamli kompetensiyalar”, “Pedagogning kasbiy kompetensiyalarini rivojlantirish” va “Astronomiya va astrofizikaning zamonaviy muammolari va yutuqlari” o'quv modullari bilan uzviy bog'langan holda pedagoglarning kasbiy pedagogik tayyorgarlik darajasini orttirishga xizmat qiladi.

Modulning oliy ta'limdagi o'rni

Modulni o'zlashtirish orqali tinglovchilar ilmiy pedagogik va ilmiy tadqiqot ishlari olib borilayotgan ta'lim muassasalarining ob'ekti, uslubiy ko'rsatmalar, yo'riqnomalar va boshqa hujjatlarni o'rganish, ilmiy tadqiqot ishlarini olib borish, ularni tahlil qilish, ilmiy tadqiqot natijalarini amaliyotga tadbqiq etish bo'yicha zaruriy bilimlar hamda kasbiy kompetentlikka ega bo'ladilar.

MODUL BO'YICHA SOATLAR TAQSIMOTI

№	Modul mavzulari	Auditoriya o'quv yuklamasi				
		Umumiy soat	Jami auditoriya soati	Jumladan		
				Nazariy	Amaliy	Ko'chma mashg'ulot
1	Model va modellashtirish tushunchasi. Modellar turlari va klassifikatsiyasi. Kompyuterda modellashtirish. Xisoblash fizikasi. Modellashtirish fizikasi. Kompyuterli eksperiment bosqichlari. Kompyuterda, modellashtirish uchun operatsion tizim. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Python tiliga kirish: proyektni saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari.	2	2	2		
2	Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish.	4	4		2	2
3	Python tiliga kirish: proyektni saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari. Kodlarni tuzish: algoritmlar, o'zgaruvchanlar, ma'lumotlar turlari. Dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollar.	2	2			2
4	Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari. Chala tarmoqli va to'liq tarmoqli algoritmlar. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash. Mantiqiy operatsiyalar. Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko'paytmalarni topish. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.	6	6	2	2	2

5	Massivlar bilan ishlash. Grafiklar chizish usullari Massivlar: bir o‘lchamli va ikki o‘lchamli massivlarni shakillantirish, elementlari bilan ishlash. Tasodidifiy jarayonlar xaqida tushuncha va tasodifiy sonlar bilan ishlash.	6	6	2	2	2
6	Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.	2	2			2
8	Integrallash hamda oddiy differensial tenglamani yechish sonli usullari. Sonli integrallash usullari va algoritmlari: to‘g‘ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari. Ular yordamida aniq integrallarni hisoblash. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari. Ekspiriment ma’lumotlarining statistika tahlili. Xarakteristik tenglamasini sonli yechish. Mexanik harakatlarni modelashtirish va ularni vizualashtirish usullari..	6	6	2	2	2
Jami:		28	28	8	8	12

NAZARIY MASHG‘ULOTLAR MAZMUNI

1-MAVZU: Model va modelashtirish tushunchasi. Modellar turlari va klassifikatsiyasi. (2 soat)

Reja:

1. Model va modelashtirish tushunchasi.
2. Modellar turlari va klassifikatsiyasi.
3. Kompyuterda modelashtirish.
4. Xisoblash fizikasi.
5. Modelashtirish fizikasi.
6. Kompyuterli eksperiment bosqichlari.
7. Kompyuterda, modelashtirish uchun operatsion tizim.

2-MAVZU: Tarmoqlanuvchi algoritmlar, if va else shart operatrlari (2 soat)

Reja

1. Tarmoqlanuvchi algoritmlar, if va else shart operatrlari bilan ishlash misollari.
2. Chala tarmoqli va to‘liq tarmoqli algoritmlar.
3. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash.
4. Mantiqiy operatsiyalar.
5. Sikllar bilan ishlash: sikllar va siklik operatorlar.
6. Sikllar yordamida summa va ko‘paytmalarni topish.
7. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.

3-MAVZU: Massivlar bilan ishlash. Grafiklar chizish usullari (2 soat)

Reja

1. Bir o'lchamli va ikki o'lchamli massivlarni shakillantirish
2. Elementlari bilan ishlash.
3. Tasodifiy jarayonlar xaqida tushuncha va tasodifiy sonlar bilan ishlash.
4. Grafiklar chizish usullari.
5. matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

4-MAVZU: Integrallash hamda oddiy differensial tenglamani yechish sonli usullari. Sonli integrallash usullari va algoritmlari (2 soat)

Reja

1. Sonli integrallash usullari algiritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.
2. Ular yordamida aniq integrallarni hisoblash.
3. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari.
4. Mexanik harakatlarni modelashtirish

AMALIY MASHG'ULOTLAR MAZMUNI

1-MAVZU: Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. (2 soat)

Reja:

1. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari.
2. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish.

2-MAVZU: Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari (2 soat)

Reja

1. Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari.
2. Chala tarmoqli va to'liq tarmoqli algoritmlar.
3. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash.
4. Mantiqiy operatsiyalar.
5. Sikllar bilan ishlash: sikllar va siklik operatorlar.
6. Sikllar yordamida summa va ko'paytmalarni topish.
7. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.

3-MAVZU: Massivlar bilan ishlash. Grafiklar chizish usullari (2 soat)

Reja:

1. Bir o'lchamli va ikki o'lchamli massivlarni shakillantirish, elementlari bilan ishlash.

2. Grafiklar chizish usullari.
3. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

4-MAVZU: Sonli integrallash usullari. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari. (2 soat)

Reja:

- 1) Sonli integrallash usullari va algoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.
- 2) Aniqlik integrallarni hisoblash.
- 3) Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari

KO'CHMA MASHG'ULOTLAR MAZMUNI

1-MAVZU: Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. (2 soat)

Reja:

3. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari.
4. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish.

2-MAVZU: Python tiliga kirish: proyektni saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari. (2 soat)

Reja

1. Python tiliga kirish: proyektni saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari.
2. Kodlarni tuzish: algoritmlar, o'zgaruvchanlar, ma'lumotlar turlari.
3. Dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollar.

3-MAVZU: Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari. (2 soat)

Reja

1. Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari.
2. Chala tarmoqli va to'liq tarmoqli algoritmlar.
3. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash.
4. Mantiqiy operatsiyalar.
5. Sikllar bilan ishlash: sikllar va siklik operatorlar.
6. Sikllar yordamida summa va ko'paytmalarni topish.
7. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.

4-MAVZU: Massivlar: bir o'lchamli va ikki o'lchamli massivlarni shakillantirish, elementlari bilan ishlash. (2 soat)

Reja

1. Bir o'lchamli va ikki o'lchamli massivlarni shakillantirish
2. Elementlari bilan ishlash.
3. Tasodidifiy jarayonlar xaqida tushuncha va tasodifiy sonlar bilan ishlash.

5-MAVZU: Grafiklar chizish usullari (2 soat)

Reja:

1. Grafiklar chizish usullari.
2. matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

6-MAVZU: Sonli integrallash usullari. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari. (2 soat)

Reja:

1. Sonli integrallash usullari va algiritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.
2. Aniq integrallarni hisoblash.
3. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari.

O'QITISH SHAKLLARI

Mazkur modulni o'qitish jarayonida ta'limning zamonaviy metodlari, pedagogik texnologiyalar va axborot-kommunikatsiya texnologiyalari qo'llanilishi nazarda tutilgan:

ma'ruza darslarida zamonaviy kompyuter texnologiyalari yordamida prezentatsion va interfaol pedagogik (Aqliy xujim, Venn diagrammasi, konseptual jadval) usul va texnologiyalardan foydalaniladi;

o'tkaziladigan amaliy mashg'ulotlarda texnik vositalardan, grafik organayzerlardan, keyslardan foydalanish, guruhli fikrlash, kichik guruhlar bilan ishlash, blis-so'rovlardan va boshqa interaktiv ta'lim usullarini qo'llash nazarda tutiladi.

II. MODULNI O‘QITISHDA FOYDALANILADIGAN INTERFAOL TA’LIM METODLARI

«FSMU» texnologiyasi

Tehnologiyaning maqsadi: Mazkur texnologiya ishtirokchilardagi umumiy fikrlardan hususiy hulosalar chiqarish, taqqoslash, qiyoslash orqali ahborotni o‘zlashtirish, hulosalash, shuningdek, mustaqil ijodiy fikrlash ko‘nikmalarini shakllantirishga xizmat qiladi. Mazkur texnologiyadan ma’ruza mashg‘ulotlarida, mustahkamlashda, o‘tilgan mavzuni so‘rashda, uyga vazifa berishda hamda amaliy mashg‘ulot natijalarini tahlil etishda foydalanish tavsiya etiladi.

Tehnologiyani amalga oshirish tartibi:

- ✓ Qatnashchilarga mavzuga oid bo‘lgan yakuniy hulosalar yoki g‘oya taklif etiladi;
- ✓ Har bir ishtirokchiga FSMU texnologiyasining bosqichlari yozilgan qog‘ozlarni tarqatiladi;
- ✓ ishtirokchilarning munosabatlari individual yoki guruhiiy tartibda taqdimot qilinadi.

FSMU tahlili qatnashchilarda kasbiy-nazariy bilimlarni amaliy mashqlar va mavjud tajribalar asosida tezroq va muvaffaqiyatli o‘zlashtirilishiga asos bo‘ladi.

F	<i>Fikringizni bayon eting</i>
S	<i>Fikringiz bayoniga sabab ko‘rsating</i>
M	<i>Ko‘rsatgan sababingizni asoslovchi misol keltiring</i>
U	<i>Fikringizni umumlashtiring</i>

“SWOT-tahlil” metodi.

Metodning maqsadi: Mavjud nazariy bilimlar va amaliy tajribalarni tahlil qilish, taqqoslash asosida muammoni hal etish yo‘llarini topishga, bilimlarni mustahkamlash, takrorlash, baholashga, mustaqil, tanqidiy fikrlashni, nostandart tafakkurni shakllantirishga mo‘ljallangan.

S – (strength)	• kuchli tomonlari
W – (weakness)	• zaif, kuchsiz tomonlari
O – (opportunity)	• imkoniyatlari
T – (threat)	• to‘siqlar

“VENN DIAGRAMMASI” STRATEGIYASI

“Venn diagrammasi” strategiyasi talabalarda mavzuga nisbatan tahliliy yondoshuv, ayrim qismlar negizida mavzuning umumiy mohiyatini o‘zlashtirish (sintezlash) ko‘nikmalarini hosil qilishga yo‘naltiriladi. Strategiya kichik guruhlarini shakllantirish asosida sxema bo‘yicha amalga oshiriladi.

Yozuv taxtasi o‘zaro teng to‘rt bo‘lakka ajratiladi va har bir bo‘lakka quyidagi sxema chiziladi:

Strategiya o‘quvchilar tomonidan o‘zlashtirilgan o‘zaro yaqin nazariy bilimlar, ma‘lumotlar yoki dalillarni qiyosiy tahlil etishga yordam beradi. Ushbu strategiyadan muayyan bo‘lim yoki boblar bo‘yicha yakuniy darslarni tashkil etishda foydalanish yanada samaralidir.

“Venn diagrammasi” strategiyani qo‘llash bosqichlari:

- ✓ tinglovchilar to‘rt guruhga bo‘linadi;
- ✓ yozuv taxtasiga topshiriqni bajarish mohiyatini aks ettiruvchi sxema chiziladi;
- ✓ har bir guruhga o‘zlashtirilayotgan mavzu (bo‘lim, bob) yuzasidan alohida topshiriqlar beriladi;
- ✓ topshiriqlar bajarilgach, guruh a‘zolari orasidan liderlar tanlanadi;
- ✓ liderlar guruh a‘zolari tomonidan bildirilgan fikrlarni umumlashtirib, yozuv taxtasida aks etgan diagrammani to‘ldiradilar

“ASSESSMENT” metodi

Metodning maqsadi: Mazkur metod ta‘lim oluvchilarning bilim darajasini baholash, nazorat qilish, o‘zlashtirish ko‘rsatkichi va amaliy ko‘nikmalarini tekshirishga yo‘naltirilgan. Mazkur texnika orqali ta‘lim oluvchilarning bilish faoliyati turli yo‘nalishlar (test, amaliy ko‘nikmalar, muammoli vaziyatlar mashqi, qiyosiy tahlil, simptomlarni aniqlash) bo‘yicha tashhis qilinadi va baholanadi.

Metodni amalga oshirish tartibi:

“Assesment” lardan ma‘ruza mashg‘ulotlarida talabalarning yoki qatnashchilarning mavjud bilim darajasini o‘rganishda, yangi ma‘lumotlarni bayon qilishda, seminar, amaliy mashg‘ulotlarda esa mavzu yoki ma‘lumotlarni o‘zlashtirish darajasini baholash, shuningdek, o‘z-o‘zini baholash maqsadida individual shaklda foydalanish tavsiya etiladi. Shuningdek, o‘qituvchining ijodiy yondashuvi hamda o‘quv maqsadlaridan kelib chiqib, assesmentga qo‘shimcha topshiriqlarni kiritish mumkin.

Topshiriq: Har bir katakdagi to‘g‘ri javob 5 ball yoki 1-5 balgacha baholang.



“KWHL” metodi

Metodning maqsadi: Mazkur metod tinglovchilarni yangi axborotlar tizimini qabul qilishi va bilimlarni tizimlashtirishi uchun qo‘llaniladi, shuningdek, bu metod tinglovchilar uchun mavzu bo‘yicha qo‘yidagi jadvalda berilgan savollarga javob topish mashqi vazifasini belgilaydi.

Izoh. KWHL:

Know – nimalarni bilaman?

Want – nimani bilishni xohlayman?

How - qanday bilib olsam bo‘ladi?

Learn - nimani o‘rganib oldim?

“KWHL” metodi	
1. Nimalarni bilaman:-	2. Nimalarni bilishni xohlayman, nimalarni bilishim kerak:-
3. Qanday qilib bilib va topib olaman:-	4. Nimalarni bilib oldim:-

“KEYS-STADI” metodi

Keys stadi 1870 yilda Garvard universitetida, keyinchalik 1920 yilda Garvard biznes maktabida qo‘llanilgan. Keyslar tipologiyasi asosiy manbalari, syujetli mavjudligi, vaziyat bayonining vaqtdagi izchilligi, keys ob‘ekti, materialni taqdim etish usuli, hajmi, tuzilmaviy xarakterga egaligi, o‘quv topshirig‘ini taqdim etish usuli, didaktik maqsadi, taqdim etish usuliga ko‘ra ajratiladi.

Keys metodini amalga oshirish bosqichlari:

- Keys bilan tanishuv;
- Asosiy muammoni ajratib olish;
- G‘oyalar yig‘ish va izlash;
- Keys yechimini tavsiya qilgan g‘oyalarni tahlil qilish;
- Keys yechimi va tavsiyalar

Asosiy tushunchalar

Keys-stadi (inglizcha *sase* - to‘plam, aniq vaziyat, *stadi* -ta’lim) - keysda bayon qilingan va ta’lim oluvchilarni muammoni ifodalash hamda uning maqsadga muvofiq tarzdagi yechimi variantlarini izlashga yo‘naltiradigan aniq real yoki sun’iy ravishda yaratilgan vaziyatning muammoli-vaziyatli tahlil etilishiga asoslanadigan *ta’lim uslubidir*.

Keys-stadi - ta’lim, axborotlar, kommunikatsiya va boshqaruvning qo‘yilgan ta’lim maqsadini amalga oshirish va keysda bayon qilingan amaliy muammoli vaziyatni hal qilish jarayonida prognoz qilinadigan o‘quv natijalariga kafolatli yetishishni vositali tarzda ta’minlaydigan bir tartibga keltirilgan optimal usullari va vositalari majmuidan iborat bo‘lgan *ta’lim texnologiyasidir*

Ish bosqichlari	Faoliyat shakli va mazmuni
1-bosqich: Keys va uning axborot ta’minoti bilan tanishtirish	□ yakka tartibdagi audio-vizual ish; □ keys bilan tanishish (matnli, audio yoki media shaklda); □ axborotni umumlashtirish; □ axborot tahlili; □ muammolarni aniqlash
2-bosqich: Keysni aniqlashtirish va o‘quv topshirig‘ni belgilash	□ individual va guruhda ishlash; □ muammolarni dolzarblik ierarxiyasini aniqlash; □ asosiy muammoli vaziyatni belgilash
3-bosqich: Keysdagi asosiy muammoni tahlil etish orqali o‘quv topshirig‘ining yechimini izlash, hal etish yo‘llarini ishlab chiqish	□ individual va guruhda ishlash; □ muqobil yechim yo‘llarini ishlab chiqish; □ har bir yechimning imkoniyatlari va to‘siqlarni tahlil qilish; □ muqobil yechimlarni tanlash
4-bosqich: Keys yechimini shakllantirish va asoslash, taqdimot.	□ yakka va guruhda ishlash; □ muqobil variantlarni amalga qo‘llash imkoniyatlarini asoslash; □ ijodiy-loyiha taqdimotini tayyorlash; □ yakuniy xulosa va vaziyat yechimining amaliy aspektlarini yoritish

NAZARIY MASHG'ULOTLAR

1-MAVZU: Model va modellashtirish tushunchasi. Modellar turlari va klassifikatsiyasi

Reja:

1. Model va modellashtirish tushunchasi.
2. Modellar turlari va klassifikatsiyasi.
3. Kompyuterda modellashtirish.
4. Xisoblash fizikasi.
5. Modellashtirish fizikasi.
6. Kompyuterli eksperiment bosqichlari.
7. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari.
8. Python tiliga kirish: proyektni saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari.

Tayanch atamalar: modellar, fizik jarayon, eksperiment, kuzatuv, analitik model, sonli model, modellashtirish bosqichlari, matematik operatorlar, mantiqiy operatorlar, resurslar, Python dasturlash tili, Matlab sistemasi, modellar turlari, algoritm, blok-sxema, chiziqli-algotirm, sintaksis.

Model va modellashtirish asoslari va tushunchalari

Modellashtirish – bu haqiqiy dunyodagi jarayonlar, tizimlar yoki hodisalarni soddalashtirilgan shaklda ifodalashdir. Modellar orqali biz murakkab tizimlarni tushunish, ularni tahlil qilish va bashorat qilish imkoniyatiga ega bo‘lamiz. Modellashtirishning asosiy maqsadi – haqiqiy jarayonlarni matematik, grafik yoki kompyuter dasturlari orqali tavsiflashdir.

Fizik jarayonlarni modellashtirish – bu tabiatdagi fizik qonuniyatlar asosida sodir bo‘ladigan hodisalarni (masalan, harakat, issiqlik uzatish, to‘lqinlar, elektr va magnit maydonlar) matematik yoki raqamli usullar bilan ifodalashdir. Bunda fizik qonunlar (masalan, Nyuton qonunlari, termodinamika qonunlari, Maksvell tenglamalari) asosiy rol o‘ynaydi. Fizika va astronomiya fanining rivojlanishida kompyuterlarning roli benihoya yuqoridir. Katta hajmdagi ma’lumotlarni tahlil qilish, bo‘layotgan hodisalar mexanizmini tushunib yetish, bo‘lishi mumkin bo‘lgan jarayonlarni oldindan aytish shular jumlasidandir. Kuzatuv ma’lumotlarini tahlili natijasida biz sodir bo‘layotgan astronomik jarayonlarni tushunishga harakat qilamiz.

Uchrashi mumkin bo'lgan har qanday masalani yechish uchun quyidagilarni bilish kerak:

- a) *qo'yilgan masalani nazariy jihatdan to'g'ri tushunish;*
- b) *masalani matematik ifodalash;*
- c) *qo'yilgan matematik masalaning sonli yechish usulini bilish;*
- d) *ushbu sonli usulda hisob-kitob olib boradigan kompyuter dasturini yaratish va uni ishga tushirish;*
- e) *olingan natijalarni to'g'ri talqin qilish.*

Fizika va astronomiyada kompyuterlar asosiy quyidagi yo'nalishlarda ishlatilishi mumkin.

Eng asosiylarini sanab o'tamiz.

- 1) *Nazariy masalalarni yechish uchun* (matematik modellashtirish va uni tenglamalar yordamida yechish);
- 2) *Ma'lumotlar bazasini yaratish* (yig'ilgan kuzatuv va nazariy ma'lumotlarni umumlashtirish va saqlash, masalan, kataloglar ko'rinishida);
- 3) *Kuzatuvlarni avtomatlashtirishda;*
- 4) *O'qitishda foydalanishda;*
- 5) *Aloqa va axborot almashish vositasi sifatida* (xalqaro kompyuterlar tarmoqlarida axborot almashinish).

Fizik jarayonlarni nimaga modellashtirish kerak?

Fizik jarayonlarni modellashtirish – bu tabiatdagi murakkab hodisalarni tushunish, bashorat qilish va ularni boshqarish uchun muhim vosita. Quyida modellashtirishning asosiy maqsadlari va zarurati haqida batafsil to'xtalib o'tamiz:

Murakkablikni soddalashtirish: Tabiatdagi ko'plab jarayonlar juda murakkab bo'lib, ularni to'liq va aniq o'rganish qiyin. Modellashtirish orqali bu jarayonlarni soddalashtirilgan shaklda ifodalash mumkin. Masalan, ob'ektning harakatini tahlil qilishda havo qarshiligini e'tiborsiz qoldirish mumkin.

Bashorat qilish va tahlil qilish: Modellar orqali kelajakda sodir bo'lishi mumkin bo'lgan hodisalarni oldindan aytish mumkin. Misol uchun, ob-havo sharoitini modellashtirish orqali yomg'ir yoki quyoshli kunlarni bashorat qilish mumkin.

Tajribalarni tejash: Ba'zi fizik jarayonlarni laboratoriya sharoitida o'rganish qimmat yoki xavfli bo'lishi mumkin. Modellashtirish orqali bu jarayonlarni kompyuterda simulyatsiya qilish mumkin. Masalan, yadro

reaksiyalarini haqiqiy tajriba qilmasdan, modellar orqali o'rganish mumkin.

Yangi texnologiyalarni ishlab chiqish: Muhandislikda yangi qurilmalar yoki tizimlarni loyihalashda modellashtirish juda muhim. Masalan, samolyotlar, avtomobillar yoki quyosh panellari ishlab chiqilishidan oldin ularning ishlash printsiplari modellar orqali tekshiriladi.

Tabiiy hodisalarni tushunish: Fizik modellar orqali tabiatdagi hodisalarni chuqurroq tushunish mumkin. Misol uchun, Yerning iqlim o'zgarishi, quyosh tizimidagi sayyoralar harakati yoki atomlar darajasidagi jarayonlar modellar yordamida o'rganiladi

Xavfsizlikni ta'minlash: Ba'zi jarayonlarni haqiqiy sharoitda o'rganish xavfli bo'lishi mumkin. Masalan, yong'in, portlash yoki yadroviy reaksiyalarni modellashtirish orqali xavfsizroq sharoitda ularni o'rganish mumkin.

Resurslarni tejash: Modellashtirish orqali real tajribalar uchun sarflanadigan vaqt, mablag' va resurslarni tejash mumkin. Masalan, qurilish loyihalarida materiallarning mustahkamligini modellashtirish orqali xatoliklarni oldindan aniqlash mumkin.

Ilmiy bilimlarni rivojlantirish: Modellashtirish orqali yangi nazariyalar va qonuniyatlarni ishlab chiqish mumkin. Masalan, kvant mexanikasi yoki nisbiylik nazariyasi kabi fundamental fizik nazariyalar modellar orqali isbotlangan.

Atrof-muhitni muhofaza qilish: Atrof-muhit muammolarini (masalan, havo ifloslanishi, suv resurslarining kamayishi) modellashtirish orqali ularni oldini olish yoki bartaraf etish choralarini ishlab chiqish mumkin.

Ta'lim va tushuntirish: Modellar orqali fizik jarayonlarni tushuntirish va o'qitish osonroq bo'ladi. Masalan, maktab va universitetlarda fizika fanini o'qitishda modellardan keng foydalaniladi.

Demak, *Fizik jarayonlarni modellashtirish* – bu nafaqat ilmiy tadqiqotlar, balki texnologiya, muhandislik, tibbiyot va atrof-muhitni muhofaza qilish kabi sohalarda muhim rol o'ynaydi. Modellar orqali biz murakkab hodisalarni tushunish, ularni bashorat qilish va samarali yechimlar topish imkoniyatiga ega bo'lamiz.

Masalan, har qanday miqyosdagi astrofizik hodisalarni kuzatish – Saturn halqalari dinamikasidan tortib, yaqin yulduzlar atrofidagi sayyoralar orbitalarigacha, uzoq o'ta yangi yulduzlar va to'qnashayotgan galaktikalardan, hatto Katta Portlashning o'zi - muqarrar ravishda bitta xulosaga olib keladi: gidrodinamik oqimlar va osmon mexanikasi Koinot

qanday ishlashini tushunish uchun mutlaqo zarur elementlari ekan.

Bundan tashqari, astrofizik jarayonlar natijasida paydo bo'lgan radiatsiya doimo Yerni deyarli sezilmaydigan miltillovchi yorug'lik bilan o'rab oladi. Bu zaif porlash voqealarning o'zi va asboblارimiz o'rtasidagi yagona bog'liqlikdir; ammo, agar u fazoviy ravishda to'g'ri taqsimlangan yoki nurlanishning turli chastotalarida parchalangan bo'lsa, u juda ko'p ma'lumotlarni o'z ichiga oladi. Shunday qilib, radiatsiya uzatish nazariyasi ushbu hodisalarni talqin qilishda uchinchi, kam bo'lmagan zarur elementni tashkil qiladi. Bundan tashqari, astrofizik ishlab chiqarish hosil bo'lgan nurlanish doimo Yerni sezilarli sezilmaydigan nur bilan o'rab oladi. Bu kuchsiz nurlanish tadqiqot etilayotgan hodisalarning o'zi va asboblارimiz o'rtasidagi yagona vositadir. Shuning uchun, agar uni fazodagi taqsimotini tushunib olsak yoki turli nurlanish chastotalariga ajrata olsak, unda juda ko'p ma'lumotlarga erishgan bo'lamiz.

Kuzatuv ma'lumotlarini nazariy talqin qilish ko'pincha bir necha bosqichlaridan o'tadi, shu jumladan, birinchi navbatda, fizikaning asosiy qonunlariga asoslangan taxminiy hisob-kitoblar; ikkinchidan, ko'pincha chiziqli g'alayonlanish nazariyasiga asoslangan analitik yoki yarim analitik hisob-kitoblar; va uchinchidan, to'liq, har tmonlama olib boriladigan raqamli modellashirish. Yangi kuzatuv imkoniyatlari kengayishi bilan astrofizik hodisalarga xos bo'lgan murakkablik tobora ravshanlashib bormoqda. Nazariy hisob-kitoblar shuni ko'rsatadiki, ayrim turdagi obyektlar va hodisalarni tushuntirishda fizika qonunlari hozircha oxirigacha ma'lum bo'lmagan holatlarga duch kelar ekanmiz va o'z navbatida ularni tushuntirish talab etiladi.

Masalan, biz shu tarzda o'rganishimiz kerak bo'lgan sohalarga yangi shakllanayotgan galaktikalar, neytron yulduzlar, ulkan sayyoralarning ichi sohalari yoki yulduzlararo muhitdagi hodisalar kirishi mumkin. Bunday ekstremal holatlarni tadqiq qilishda, kuzatuv ma'lumotlardan tashqari, gidrodinamika va nur o'tkazish qonunlari, kimyo qonunlari, yadro fizikasi, magnit maydonlari va atom energiyasi, modda xususiyatlari, holat tenglamasi kabi vaziyatlarni inobatga olinishini talab etadi. Fizik effektlarning keng doirasi ba'zi holatlarda ancha murakkab modellarni tuzish va o'zragnishni talab qiladi.

Modellashtirish turlari, klassifikatsiyasi

Analitik modellashtirish: Matematik tenglamalarni aniq yechish orqali amalga oshiriladi. Bu usul soddaroq tizimlar uchun qo'llaniladi.

Raqamli modellash: Murakkab tizimlarda differensial tenglamalarni raqamli usullar (masalan, chekli farqlar usuli) bilan yechish.

Kompyuter modellash: Fizik jarayonlarni kompyuter dasturlari yordamida simulyatsiya qilish (masalan, Monte-Karlo usuli, molekulyar dinamika).

Analitik usullar doimo fizika qonunlariga yaqinlashishni o'z ichiga oladi, ammo vaziyatga qarab bunday taxminlar oqilona yoki yo'q bo'lishi mumkin. Ba'zi hollarda inobatga olinishi kerak bo'lgan jarayonlarning xilma-xilligi umumiy muammoni sof analitik usullar yordamida hal qilishga imkon bermaydi. Bunday usullar, albatta, muammoning alohida qismlarini shakllantirish uchun zarurdir, ammo shunday vaziyatlar tug'iladiki, analitik usullar ularni yecha olmaydi va natijada sonli yondashuvga murojaat qilish kerak bo'ladi.

Kompyuter quvvatining jadal rivojlanishi va ma'lumotlarni saqlash tizimlarining mavjudligi bilan bunday modellash tobora murakkablashmoqda, natijalarning grafik imkoniyatlari esa ajoyib tasvirlarga olib keldi.

Aytilgan masalalarni yechish uchun talabalardan dasturlash asoslarini bilish va biron bir dasturlash tili bilan ishlash ko'nikmalari talab etiladi. Shundan kelib chiqqan holda, bu o'quv qo'llanmaning asosiy maqsadi – astronomiya sohasida uchraydigan masalalarni sonli yechish orqali Matlab yoki unga alternativ bo'lgan Octave dasturlash muhitidan foydalanishni, asosiy sonli usullarni o'rgatishdan iborat.

Fizik yoki astronomik masalasini kompyuter yordamida yechilishi ko'pincha sonli modellash masalasiga taqaladi. Umuman olganda kompyuterdagi modellash - bu kompyuter modelini (yoxud sonli, raqamli modelni) bir yoki bir nechta hisoblash tugunlarida hisoblash jarayoni deb tushuniladi. Shunda obyekt, tizim yoki yondoshishni haqiqiydan farq qiladigan, ammo algoritmik nuqtayi nazaridan yaqinlashtirilgan shaklda aks etiladi. U tizimning xususiyatlarini va vaqt o'tishi bilan ularning o'zgarishi dinamikasini tavsiflovchi ma'lumotlarni o'z ichiga oladi.

Kompyuter modellari umumiy matematik modellash vositasiga aylandi va fizika, astrofizika, mexanika, kimyo, biologiya, iqtisodiyot, sotsiologiya, meteorologiya, boshqa fanlarda hamda radioelektronika, mashinasozlik, avtomobilsozlik va shu kabi texnikaning turli sohalarida amaliy vazifalarni yechishda qo'llaniladi. Kompyuter modellari obyekt haqida yangi bilimlarni olish yoki analitik tarzda yechib bo'lmaydigan

juda murakkab tizimlarning xatti-harakatlarini taxmin qilish uchun ishlatiladi.

Kompyuter modellashtirish murakkab tizimlarni o'rganishning eng samarali usullaridan biridir. Haqiqiy tajribalar moliyaviy yoki tabiiy to'siqlar tufayli qiyin bo'lgan yoki oldindan aytib bo'lmaydigan natijalar berishi mumkin bo'lgan hollarda hisoblash tajribalarini o'tkazish imkoniyati tufayli kompyuter modellarini o'rganish osonroq va qulayroqdir. Kompyuter modellarini rasmiylashtirilganligi o'rganilayotgan asl obyektning (yoki obyektlar sinfining) xususiyatlarini belgilovchi asosiy omillarni aniqlashga imkon beradi. Xususan, modellashtirilayotgan fizik sistemaning uning parametrlari va dastlabki sharoitlarining o'zgarishiga ta'sir qilishi o'rganiladi.

Kompyuter modelini yaratishda o'rganilayotgan obyekt yoki hodisaning xos tabiatidan abstraksiyalanishga asoslanadi va ikki bosqichdan iborat – avval sifatiy, so'ngra sonli model yaratiladi. Qanchalik muhim xususiyatlar aniqlansa va kompyuter modeliga o'tkazilsa, u haqiqiy modelga shunchalik yaqin bo'ladi, va natijada ushbu modeldan foydalanadigan tizim shunchalik ko'p imkoniyatlarga ega bo'lishi mumkin.

Sonli modellashtirish kompyuterda bir qator hisoblash tajribalarini o'tkazishdan iborat bo'lib, ularning maqsadi hisob-kitob natijalarini tahlil qilish, talqin qilish va o'rganilayotgan obyektning haqiqiy xatti-harakatlari bilan taqqoslash va agar kerak bo'lsa, modelni yanada takomillashtirishga yo'naltirgan.

Analitik va imitatsion modellashtirish mavjud. Analitik modellashtirishda haqiqiy obyektning matematik, ya'ni abstrakt modellari algebraik, differensial va boshqa tenglamalar shaklida o'rganiladi, shuningdek ularning aniq yechimiga olib keladigan aniq hisoblash usullarining amalga oshirilishi ta'minlaydi. Imitatsion modellashtirishda esa matematik modellar ko'p sonli elementar operatsiyalarni ketma-ket bajarish orqali o'rganilayotgan obyekt yoki tizimning ishlashini takrorlaydigan algoritmlar shaklida o'rganiladi.

Fizik modellashtirishning amaliy ahamiyati kelsak, u qatir amaliy masalalarni hal etish yordam beradi.

Masalan:

Ilmiy tadqiqotlar: Fizik hodisalarni chuqurroq tushunish.

Muhandislik: Qurilmalar va tizimlarni loyihalash (masalan, samolyotlar, quyosh panellari).

Tibbiyot: Tana ichidagi fizik jarayonlarni o'rganish (masalan, qon

oqimi, nafas olish).

Atrof-muhitni muhofaza qilish: Iqlim o'zgarishi, havo va suv oqimlarini modellashtirish.

Misollar: Erkin tushish harakati: Nyuton qonunlari asosida ob'ektning erkin tushishini tavsiflovchi matematik model. Issiqlik uzatish: Issiqlik o'tkazuvchanlik tenglamasi yordamida materiallarda issiqlik tarqalishini modellashtirish. Suyuqlik dinamikasi: Navye-Stoks tenglamalari yordamida suyuqlik oqimini tahlil qilish.

Umuman olganda, *fizik jarayonlarni modellashtirish* – bu tabiat qonuniyatlarini tushunish va ularni amaliyotda qo'llashning muhim vositasi. U ilmiy tadqiqotlar, muhandislik va texnologiyalarning rivojlanishida asosiy rol o'ynaydi.

Kompyuterli modellashtirish quyidagilarga imkonini beradi:

- tadqiqot obyektlari doirasini kengaytirish, ya'ni takrorlanmaydigan hodisalarni, ko'zdan yashiringan, o'tmish va kelajakdagi hodisalarini, real sharoitda ro'yobga chiqmaydigan obyekt va hodisalarni o'rganish;

- har qanday tabiatdagi obyektlarni, shu jumladan mavhumlarni, tasavvur qilish;

- hodisa va jarayonlarni yuz berish dinamikasida o'rganish;

- vaqtni boshqarish (tezlashtirish, sekinlashtirish va hk.);

- modelning ko'p sonli sinovlarini o'tkazish, har safar uni boshlang'ich holatiga qaytarish;

- obyektning turli xil xususiyatlarini raqamli yoki grafik shaklda olish;

- obyektning sinov nusxalarini yaratmasdan uning optimal tuzilishini aniqlash;

- inson salomatligi yoki atrof-muhit uchun salbiy oqibatlarga olib kelmaydigan, xavfsiz tajribalar o'tkazish.

Shunda kompyuter modellashtirishning asosiy bosqichlari odatda quyidagilardan iborat:

- ✓ Masalani qo'yish va uni tahlil qilish, ya'ni modelni yaratish maqsadini va qanday dastlabki ma'lumotlar kerakligini aniqlash kerak.

- ✓ Modelning parametrlari o'rtasidagi bog'lanishlarni aniqlash va tavsiflash; ularning qaysi biri ushbu vazifa uchun eng muhim ekanligini va qaysi birini e'tiborsiz qoldirish mumkinligini baholash.

- ✓ Kompyuter modelini amalga oshirish usuli va algoritmini ishlab chiqish: natijalarni olish usulini tanlash yoki ishlab chiqish; tanlangan usul asosida natijalarni olish algoritmini yaratish; algoritmnining

to'g'riligini tekshirish.

✓ Kompyuter modelini (dasturini) ishlab chiqish: algoritmni kompyuterda dasturiy ta'minotni amalga oshirish vositalarini (dasturlash tili, muhiti, o'pretsion sistemasi va hokazo) tanlash; kompyuter modelini ishlab chiqish; yaratilgan dasturining to'g'riligini tekshirish.

✓ Tajriba o'tkazish: yaratilgan kompyuter modeli asosida tajriba o'tkazish; natijalarni tahlil qilish; model va obyekt xususiyatlari to'g'risida xulosa chiqarish.

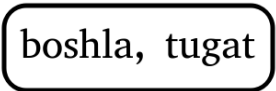
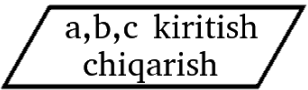
Python tiliga kirish: asosi qoidalari. Kodlarni tuzish

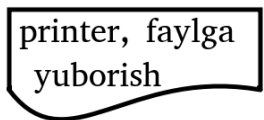
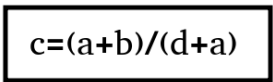
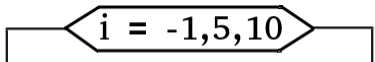
Ko'rib turganimizdek, kompyuter modellashtirishning asosiy masala algoritmgaga taqaladi. Algoritm so'zi buyuk ajdodimiz Al-Xorazmiy ismidan kelib chiqadi. Uning ismi o'rta asrlarda yevropa tillariga tarjima qilinganda algoritm so'ziga aylandi. Xozirda algoritmning ko'p sonli ta'riflari mavjud, chunki bu tushuncha keng ma'noli bo'lib, ilm, fan va kunadlik xayotda qo'llaniladi.

Algoritm deb biz ma'lum bir masalani yechishga mo'ljallangan aniq va ravon bayon etilgan harakat va amallar ketma-ketligini tushunamiz. Algoritmning mohiyati bosqichma-bosqich harakatlar ketma-ketligida bo'lib, ularning belgilangan tartibda bajarilishi qo'yilgan masalani muvaffaqiyatli yechishga olib keladi.

Ishlab chiqilgan algoritm yoki dastur to'g'ri ishlashi uchun qator xususiyatlarga ega bo'lishi kerak: diskretlik – algoritm elementar qadamlar – buyruqlarga bo'linishi kerak; bir qiymatlilik – har bir buyruq ijrochining mumkin bo'lgan yagona harakatini belgilaydi; tushunarliklik – algoritmning barcha buyruqlari ijrochining buyruqlar tizimiga kiritilgan; natijalilik – ijrochi qo'yilgan masalani soni cheklangan qadamlar davomida ichishi kerak.

Algoritmning ko'pchiligi umumiylik hususiyatiga ega bo'lishi kerak – ishlab chiqilgan algoritm yordamida qator boshqa o'xshash masalalarni yechish mumkin bo'lishi kerak.

1-Jadval. Blok-sxemaning asosiy elementlari	
 boshla, tugat	Algoritmning boshlanishi va tugashi, dasturga kirish va chiqish
 a,b,c kiritish chiqarish	Vvod-vivod v obshem vide

	Natijalarni printerda chop etish yoki faylga yozish
	Jarayon: hisoblash amallari yoki buyruqlar ketma-ketligi
	Hal qilish, shartni tekshirish
	Modifikatsiya: siklni boshlash

Algoritm turli yo‘llar bilan ifodalanishi mumkin. Masalan, algoritmnı og‘zaki yoki yozma ravishda bayon qilsa bo‘ladi. Ushbu shaklda biz retseptlar, ko‘rsatmalar, instruksiyalar va boshqalardan foydalanamiz. Rasmiy ijrochilar uchun mo‘ljallangan algoritmlarnı yozish uchun maxsus dasturlash tillari ishlab chiqilgan. Bulardan tashqari algoritmlarnı grafik ko‘rinishda blok-sxema yordamida tasvirlash ham mumkin. Buning uchun maxsus belgilash tizimi ishlab chiqilgan. Blok sxemalardagi asosiy elementlari 1- jadvalda ko‘rsatilgan.

Algoritmlarning turlari xilma-xil bo‘ladi. Ularning eng asosiylarini sanab o‘tadigan bo‘lsak, avvalo chiziqli algoritmnı aytamiz. Unda har bir buyruq va instruksiya berilgan ketma-ketlikda bir marta bajariladi. Bu eng sodda algoritm bo‘lib, unig yordamida, masalan oddiy chiziqli tenglamalar yechilishi mumkin. Undan keyin tarmoqlanuvchi algoritm keladi. Unda oldindan berilgan shartning bajarilishiga qarab (to‘g‘ri yoki yolg‘on), hisoblash faqat bir marta bajariladigan turli tarmoqlar bo‘ylab davom etiladi. Keyin takrorlanuvchi algoritm keladi, unda biron bir shart tekshirilishi yoki berilgan parametrning har xil qiymatlari asosida algoritmda takrorlanish yuz beradigan jarayon. Albatta, asli masalalarnı yechganda bittasida to‘xtamasdan, algoritmlarning turli kombinatsiyalaridan foydalanamiz va bunisi aralash (kombinatsiyalangan) algoritm deyiladi. Keyingi paragraflarda bu algoritmlarning har birida batafsilroq to‘xtab o‘tamiz.

Masalalarnı yechish hamda algoritmlarnı amalga oshirish uchun dasturlash tillarning birontasini tanlab olish hamda uni o‘zlashtirish talab etiladi. Bu borada biz hozirgi paytda juda yoqori rivoj hamda mashhur bo‘lib qolgan Python dasturlash tilini taklif qilamiz.

Python nima?

Python – bu yuqori darajadagi, umummaqsadli va o‘qish oson dasturlash tili. Uning sintaksisi oddiy va qisqa bo‘lib, yangi boshlanuvchilar uchun juda qulay. Python dasturlash tili web-ilovalar, ma’lumotlar tahlili, sun’iy intellekt, ilmiy hisoblash va boshqa ko‘plab sohalarda qo‘llaniladi.

Pythonning afzalliklari:

Oson o‘rganish: Sintaksisi oddiy va ingliz tiliga o‘xshash.

Ko‘p maqsadli: Web, ilmiy hisoblash, ma’lumotlar tahlili, sun’iy intellekt kabi sohalarda qo‘llaniladi.

Keng kutubxonalar: NumPy, Pandas, TensorFlow kabi kutubxonalar orqali murakkab vazifalarni oson bajarish mumkin.

Platformadan mustaqil: Windows, macOS, Linux kabi operatsion tizimlarda ishlaydi

Python dasturlash tilining mohiyati.

Python – bu yuqori darajali, interpretatsiya qilinadigan va umummaqsadli dasturlash tili. Uning asosiy xususiyatlari quyidagilardan iborat:

Oddiy va o‘qish oson sintaksis: Python kodlari ingliz tiliga o‘xshash va kam belgilar bilan yoziladi. Bu uni yangi boshlanuvchilar uchun juda qulay qiladi.

Interpretatsiya qilinadigan til: Python kodini kompilyatsiya qilish shart emas. Kod to‘g‘ridan-to‘g‘ri interpretator orqali ishga tushiriladi.

Dinamik tipizatsiya: O‘zgaruvchilarning turini aniq ko‘rsatish shart emas. Misol uchun, $x = 10$ yozish kifoya, x butun son ekanligini Python o‘zi aniqlaydi.

Keng kutubxonalar: Python-da ma’lumotlar tahlili, sun’iy intellekt, veb-ilovalar va boshqa sohalar uchun ko‘plab kutubxonalar mavjud.

Platformadan mustaqil: Python kodini Windows, macOS, Linux kabi har qanday operatsion tizimda ishlatish mumkin.

Python-ning rivojlanishi tarixi

Python dasturlash tili Guido van Rossum tomonidan 1980-yillarning oxirida ishlab chiqilgan. Ushbu tilning yaratilishida asosiy maqsad – bu dasturchilar uchun oddiy, o‘qish oson va funksional til yaratish edi. Python nomi Guido van Rossumning sevimli komediya shousi "Monty Python's Flying Circus" dan olingan.

Asosiy bosqichlar:

1980-yillar: Guido van Rossum ABC tiliga o'xshash, lekin undan yaxshiroq til yaratish niyatida ish boshladi.

1991-yil: Pythonning birinchi versiyasi (0.9.0) chiqdi. Bu versiyada klasslar, funktsiyalar va asosiy ma'lumot turlari mavjud edi.

2000-yil: Python 2.0 chiqdi. Ushbu versiyada yangi xususiyatlar, masalan, garbage collection (axlat yig'ish) va Unicode qo'llab-quvvatlandi.

2008-yil: Python 3.0 chiqdi. Bu versiyada tilning sintaksis va imkoniyatlarida katta o'zgarishlar amalga oshirildi. Python 2.x va Python 3.x versiyalari bir-biri bilan to'liq mos kelmaydi.

Hozirgi kunda: Python dunyodagi eng mashhur dasturlash tillaridan biri hisoblanadi. U ma'lumotlar tahlili, sun'iy intellekt, veb-ilovalar, ilmiy hisoblash va boshqa sohalarda keng qo'llaniladi.

Python nima uchun qulay?

Oson o'rganish: Sintaksisi oddiy va mantiqiy.

Keng jamiyat: Python dasturchilar jamoasi katta bo'lib, har qanday muammoga yechim topish oson.

Ko'p maqsadli: Turli sohalarda qo'llash mumkin.

Kutubxonalar: NumPy, Pandas, TensorFlow, Django kabi kutubxonalar orqali murakkab vazifalarni oson bajarish mumkin.

Python dasturlashni boshlash uchun zarur vositalar:

Python interpretatori: Python kodini ishga tushirish uchun kompyuteringizda Python interpretatori o'rnatilgan bo'lishi kerak. Rasmiy sayt: python.org.

Matn muharriri yoki IDE: Kod yozish uchun oddiy matn muharriri (masalan, Notepad++) yoki maxsus IDE (masalan, PyCharm, VS Code) ishlatishingiz mumkin.

4. Python dasturining asosiy qismlari:

1. **Print funksiyasi:** Ekranga ma'lumot chiqarish uchun ishlatiladi.

```
python  
print("Salom, Dunyo!")
```

Natija: Salom, Dunyo!

O'zgaruvchilar: Ma'lumotlarni saqlash uchun ishlatiladi. Python-da o'zgaruvchilar turini aniq ko'rsatish shart emas.

```
python  
ism = "Ali"  
yosh = 25
```

```
print(ism, yosh)
```

Natija: Ali 25

Ma'lumot turlari:

String (Matn): "Salom"

Integer (Butun son): 10

Float (Haqiqiy son): 3.14

Boolean (Mantiqiy qiymat): True yoki False

List (Ro'yxat): [1, 2, 3]

Dictionary (Lug'at): {"ism": "Ali", "yosh": 25}

Shart operatorlari (if-else): Shartga qarab kod bloklarini boshqarish.

```
python
```

```
yosh = 18
```

```
if yosh >= 18:
```

```
    print("Siz voyaga yetgansiz!")
```

```
else:
```

```
    print("Siz hali yoshsiz!")
```

Natija: Siz voyaga yetgansiz!

Sikllar (for, while): Bir necha marta takrorlanadigan amallarni bajarish.

For sikli:

```
python
```

```
for i in range(5):
```

```
    print(i)
```

Natija: 0 1 2 3 4

While sikli:

```
python
```

```
son = 0
```

```
while son < 5:
```

```
    print(son)
```

```
    son += 1
```

Natija: 0 1 2 3 4

Funksiyalar: Kodni qayta ishlatish uchun funksiyalar yaratish mumkin.

```
python
```

```
def salom_ber(ism):
```

```
    print(f"Salom, {ism}!")
```

```
salom_ber("Ali")
```

Natija: Salom, Ali!

Python-ning asosiy sintaksis qoidalari:

- **Indentatsiya (joy tashlash):** Python-da kod bloklari indentatsiya orqali ajratiladi. Boshqa tillardagi `{}` qavslar o'rniga, Python joy tashlashdan foydalanadi.
- **Izohlar:** Kodga izoh qo'shish uchun `#` belgisi ishlatiladi.

```
python
# Bu izoh
print("Salom!")
```

- **Qatorni tugatish:** Python-da har bir buyruq yangi qatorda yoziladi. Agar qatorni uzun bo'lsa, `\` belgisi yordamida davom ettirish mumkin.

Python kutubxonalari:

Python-da kutubxonalar orqali murakkab vazifalarni oson bajarish mumkin. Ba'zi mashhur kutubxonalar:

- **NumPy:** Raqamli hisoblashlar uchun.
- **Pandas:** Ma'lumotlar tahlili uchun.
- **Matplotlib:** Grafiklar chizish uchun.
- **TensorFlow:** Sun'iy intellekt va mashina o'rganishi uchun.

Python dasturini ishga tushirish:

Python dasturini `.py` kengaytmali fayl sifatida saqlang va terminalda quyidagi buyruq orqali ishga tushiring:

```
bash
python dastur_nomi.py
```

Python – bu kuchli va oson o'rganiladigan dasturlash tili bo'lib, u yordamida siz turli xil dasturlar yaratishingiz mumkin. Boshlang'ich qadamlarni o'rganib, asta-sekin murakkab loyihalarga o'tishingiz mumkin.

Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari

MATLAB so'zi MATrix LABoratory – matritsa laboratoriyasi so'zlarining boshlang'ich harflaridan iborat. Bu sistemaning nomi uning mohiyatini to'liq aks ettiradi. Bu haqiqatan ham matritsa laboratoriyasi, bu yerda boshlang'ich element bu oddiy o'zgaruvchi yoki konstanta emas, balki matritsa va uning maxsus holatlari — vektor-satr, vektor-ustundir. MATLABdan keng ko'lamli ilmiy va muhandislik muammolarini hal qilish uchun foydalansa bo'ladi. Bunday vazifalarni yechishdagi hisob-kitoblarning raqamli natijalari mos ravishda vizualizatsiyasi bilan birga ko'rsatilishi kerak. MATLAB bunday muammoning ham yechimini bera oladi, bu esa o'quv jarayoni uchun juda foydali, chunki talabalar

o'zlarining kurs va bitiruv ishlarining grafik qismini tezda tayyorlashlari kerak bo'lganda bu ko'nikmalar juda muhim rol o'ynaydi. Matlabdan tashqari, albatta, Visual Digital Fortran, Delphi, Visual C++, C# va boshqalar kabi juda kuchli dasturlash va vizualizatsiyalash vositalari mavjud. Biroq, mualliflarning fikriga ko'ra, MATLAB muhiti eng sodda va ayni paytda samarali dasturlar va imkoniyatlarni beradi.

Foydalanuvchi MATLAB tizimi bilan, boshqa har qanday matematik tizim singari, klaviatura va sichqoncha yordamida aloqa o'rnatadi. Operatsion dasturlari bo'lgan tegishli belgilarni kiritish va tugmachalarni bosish orqali foydalanuvchi ma'lumotlarni kiritadi, funksiyalar va buyruqlarga murojaat qiladi va yechim oladi.

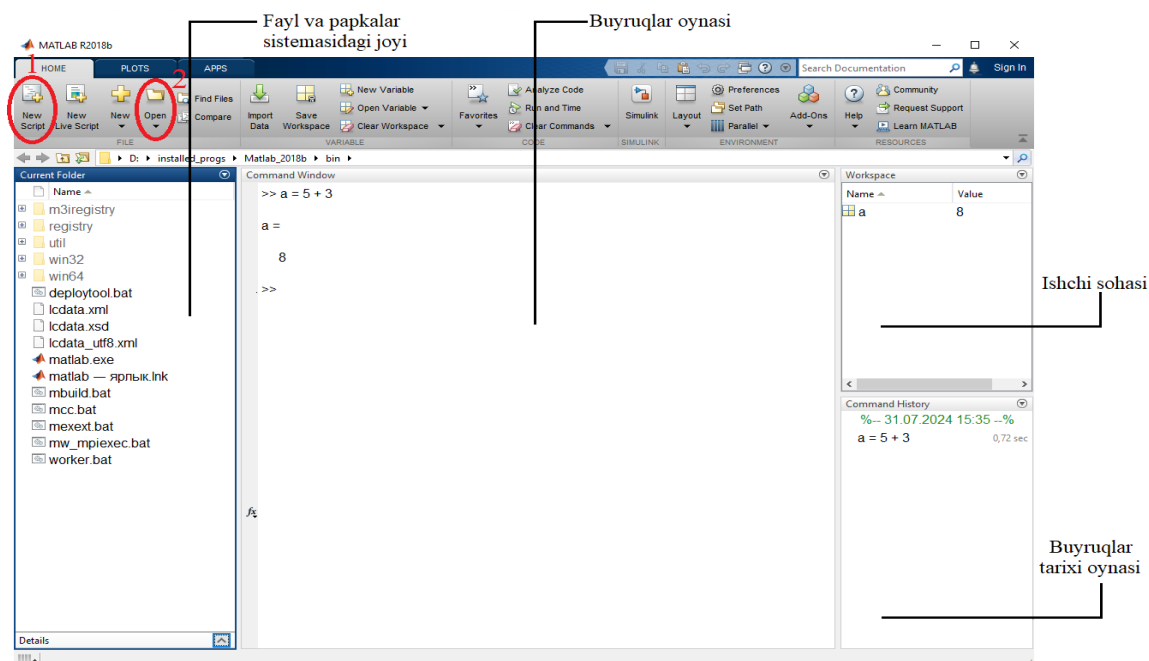
Keng foydalanuvchi interfeysi ko'plab operatsiyalarni o'z ichiga oladi va ko'p sonli funksiyalar, buyruqlar va tugmachalarni bilishni (aniqrog'i yodlashni) talab qiladi. Bularga, xususan: sistemaning asosiy menyusi; instrumentlar panelidagi tugmalar; tizim oynalari; fayl muharriri; ilovalar bilan aloqa tuzish; grafik tizimi; ma'lumotlar tizimi va boshqalar. (Bu qo'llanmada sistema va tizim so'zlari sinonimlar bo'lib, bir biriga teng).

Faqat masala va vazifalarni yechish jarayonidagina bularning barchasini o'rganib, faol foydalanish ko'nikmasini hosil qilish mumkin. MATLAB tizim bilan ishlashning boshida interfeys elementlarining hammasini to'liq o'rganish mantiqqa ega emas. Ushbu qo'llanmada sistemaning faqat dastlabki asosiy oynalari, menyu va instrumentlar paneli qisqacha tushuntirilgan.

MATLAB bilan ishlashni boshlaymiz. Odatda bu tizimni ishga tushirish uchun kompyuterning Ish Stolida joylashgan belgiga sichqonchaning chap tugmasini ikki marta bosamiz. Kompyuterning ichki imkoniyatlariga qarab, tez orada 2-rasmda ko'rsatilgan MATLAB interfeysi yuklanadi.

Foydalanuvchi tizim bilan maxsus oynalar orqali aloqa qiladi. Ularni qisqacha ta'riflaymiz.

1-Rasm. MATLAB sistemasining asosiy oynasi va sohalari



Buyruqlar oynasi (Command Window). Buyruqlar oynasi (1-rasm) foydalanuvchi uchun eng muhim. Shu oyna orqali matematik ifodalar kiritiladi hamda sistema yubormoqchi bo'lgan ma'lumotlar aks etadi. Dastur ishga tushirilgandan so'ng foydalanuvchi darhol u bilan ishlashi mumkin bo'ladi. Matematik ifodalar buyruqlar satrida taklif etuvchi `>>` belgisidan keyin yoziladi.

Kiritish satrida quyidagi ifodani kiritamiz (kiritish satrida yozilishi kerak bo'lgan misollar asosiy matndan alohida shifr bilan ajratilgan)

`>> x=2+3`

Amalni bajarish uchun Enter tugmani bosing. Natijani 1-rasmda ko'rish mumkin. Agar biz, masalan, raqamlardan birini o'zgartirmoqchi bo'lsak, buni qila olmaymiz. Kursorni istalgan qatorga qo'yish orqali avval kiritilgan buyruqni tahrirlashning iloji yo'qligi MATLAB tizimining xususiyatlaridan biridir. Avval kiritilgan buyruqni taxrirlash uchun kursorni kiritish satriga olib kelib `<↑>` va `<↓>` klavishalaridan foydalanamiz. Bu tugmalar yordamida avval kiritilgan buyruqlar ro'yxatidagi amallarni birma-bir chaqirtirib, kerakligida to'xmash kerak bo'ladi. Buyruqni darhol yoki taxrirlangandan keyin ishga tushirsa bo'ladi (eslatma: har qanaqa buyruqni ijro etish uchun Enter klavishasini bosamiz).

Ushbu oyna asosiy bo'lib, umumay holda, MATLAB sistemasi bilan ishlaganda boshqa oynalar kerak bo'lmasligi ham mumkin. Ammo ularning mavjudligi kompyuter bilan muloqotni oson va qulay qiladi.

2. Ishchi sohasi (Workspace) oynasi. Dastur bajarilishi davomida turli tipdagi o'zgaruvchanlar ishlatiladi. Hosil qilingan o'zgaruvchanlar kompyuter xotirasining maxsus ajratilgan sohasida saqlanadi. Ular o'zidan o'zi yo'q bo'lib ketmaydi – faqat lasturdan chiqqanda yoki maxsus buyruqlar yordamida. Shunda o'zgaruvchanlarni (aniqrog'i qiymatlarini) kiritilayotgan ixtiyoriy matematik ifodada ishlatib bo'ladi. Ishchi sohasi oynasi foydalanuvchiga ishchi sohasida saqlanayotgan barcha o'zgaruvchanlar ro'yxitini taqdim etadi (1-rasm). Ixtiyoriy o'zgaruvchanni tanlasa, qiymatini ko'rsa yoki uni ustida biron bir boshqa amalni bajarsa bo'ladi. Ko'rsatilgan harakatlar kontekst menyusi orqali amalga oshiriladi. Shunda sichqonchaning o'ng tugmasini o'zgaruvchi nomiga bosish kerak bo'ladi.

3. Current Folder oynasi foydalanuvchining fayllar va papkalar sistemasidagi joyini ko'rsatadi. Buyruqlar oynasidan kiritilgan matematik ifodalardan tashqari foydalanuvchi fayllar bilan ham ishlashi mumkin (ular haqida keyinroq batafsil ma'lumot beramiz). Shuningdek, matematik funksiyalar aslida shu funksiyaning nomi bilan yozilgan fayldir. Bu fayllarda berilgan funksiyani amalga oshiradigan dasturlar yozilgan. Shunday qilib, foydalanuvchi doimo fayllar bilan ish tutadi. Masalan, o'rnatilgan funksiyani ko'rsatganimizda, biz aslida shu funksiyaning dasutri matni yozilgan faylga (kengaytmasiz) murojat qilgan bo'lamiz. Shunda sistema so'ralgan faylni joriy katalogda yoki murojat etish yo'lida qidiradi.

4. Buyruqlar tarixi (Command History) oynasi. Buyruqlar oynasining kiritish satrida yozilgan barcha buyruqlar avtomatik ravishda ro'yxatga kiritiladi va u buyruqlar tarixi oynasida ko'rsatiladi. Bu ro'yxat nimaga kerak bo'lishi mumkin? Ancha avval bajarilgan buyruqqa murojaat qilish kerak bo'lib qolsa, uni aynan buyruqlar tarixi oynasidan topsa bo'ladi va unga sichqonchaning chap tugmasini ikki marta bosib, buyruq qayta ishga tushiriladi.

Buyruqlar oynasida kiritish satrida buyruqlarning kerakli ketma-ketligini bajarsak, shunga mos ketma-ketlik buyruqlar tarixi oynasida ko'rsatiladi. Bir nechta amallar yordamida buyruqlar ketma-ketligi dasturga aylanishi mumkin. Ushbu oynadagi ma'lumotlar MATLAB sistemasidan chiqib ketsak ham saqlanib boradi. Uni faqat menyu orqali tozalasa bo'ladi.

Xuddi shunga o'xshash amallarni Spyder muhitida bajarish mumkin

Spyder (Scientific PYthon Development EnviRonment) — bu **Python dasturlash muhitlaridan biri** bo'lib, ayniqsa ilmiy hisob-

kitoblar, ma'lumotlar tahlili va muhandislik sohalari uchun juda qulay vosita hisoblanadi. U **MATLAB** uslubidagi interfeysga ega bo'lib, ilmiy dasturchilar va tadqiqotchilar uchun qulay muhit yaratadi.

Spyder-ning asosiy xususiyatlari

Interaktiv konsol – Jupyter Kernel bilan bog'langan bo'lib, kodni qatorma-qator yoki blokma-blok bajarish imkonini beradi.

O'zgaruvchilar menejeri (Variable Explorer) – Matritsa, massiv va boshqa o'zgaruvchilarni vizual ravishda ko'rish va tahrirlash mumkin (MATLAB-ning "Workspace" ga o'xshash).

Matn muharriri (Editor) – Pythonda kod yozish uchun qulay muharrir bo'lib, kodni rangli qilib ko'rsatadi, auto-tugallash funksiyasiga ega va hatolarni aniqlashda yordam beradi.

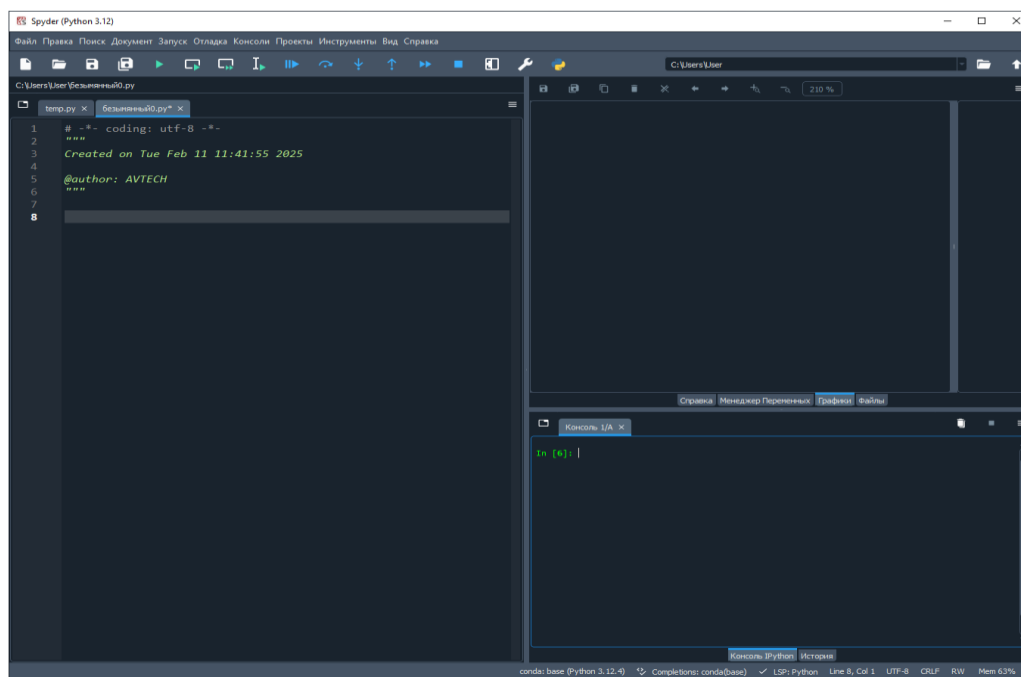
Plot va grafikalar – Matplotlib va Seaborn kabi kutubxonalar bilan yaxshi integratsiyalashgan bo'lib, grafiklarni ko'rsatishda qulay.

Debug vositasi – Koddagi xatolarni topish va tuzatish uchun kuchli debugging funksiyasi mavjud.

To'g'ridan-to'g'ri ma'lumotlarni yuklash va tahlil qilish – CSV, Excel va boshqa formatlardagi ma'lumotlarni yuklab, tahlil qilish mumkin.

2-Rasm. Spyder muhitining asosiy oynasi va sohalari

Matn muharriri (Editor)



**Plot va
grafikalar**

O'zgaruvchilar menejeri (Konsol)

Endi konsolning satrida matematik operatsiyalarni bajarishimiz mumkin. Uning uchun avvalo arifmetik operatorlarini keltiramiz. Arifmetik operatorlar. Bu operatorlar sonlar ustida asosiy matematik amallarni bajarish uchun ishlatiladi.

Operator	Tavsifi	Misol	Natija
+	Qo'shish	$5 + 3$	8
-	Ayirish	$10 - 4$	6
*	Ko'paytirish	$6 * 7$	42
/	Bo'lish (haqiqiy)	$9 / 2$	4.5
//	Bo'lish (butun qismi)	$9 // 2$	4
%	Qoldiq olish	$9 \% 2$	1
**	Darajaga oshirish	$3 ** 2$	9

Belgilash (assign) operatorlari. Bu operatorlar o'zgaruvchilarga qiymat berish yoki o'zgaruvchilarning hozirgi qiymatini o'zgartirish uchun ishlatiladi.

Operator	Tavsifi	Misol	Teng natija
=	O'zgaruvchiga qiymat berish	$x = 5$	$x = 5$
+=	Qo'shish va berish	$x += 3$	$x = x + 3$
-=	Ayirish va berish	$x -= 2$	$x = x - 2$
*=	Ko'paytirish va berish	$x *= 4$	$x = x * 4$
/=	Bo'lish va berish	$x /= 2$	$x = x / 2$
//=	Butun bo'lish va berish	$x //= 3$	$x = x // 3$
%=	Qoldiq olish va berish	$x \% = 3$	$x = x \% 3$
**=	Daraja va berish	$x ** = 2$	$x = x ** 2$

Python-dagi arifmetik operatorlar – bu sonlar ustida arifmetik amallar bajarish uchun ishlatiladigan operatorlardir. Quyida ularning ishlash usuli va misollar berilgan:

Qo'shish (+). Bu operator ikkita sonni qo'shish uchun ishlatiladi.

`a = 10`

`b = 5`

`c = a + b`

`print(c)` # Natija: 15

Qo'shish operatori satr (string) birlashtirishda ham ishlaydi

```
s1 = "Assalomu "  
s2 = "alaykum"  
print(s1 + s2) # Natija: Assalomu alaykum
```

Ayirish (-). Bu operator bir sonni ikkinchisidan ayirish uchun ishlatiladi

```
a = 10  
b = 3  
c = a - b  
print(c) # Natija: 7
```

Agar o'zgaruvchiga manfiy son bersak:

```
x = -5  
y = 3  
print(x - y) # Natija: -8
```

Ko'paytirish (*). Bu operator ikki sonni ko'paytiradi.

```
a = 6  
b = 7  
c = a * b  
print(c) # Natija: 42
```

Satlarga nisbatan ishlatilsa, takrorlash uchun foydalaniladi:

```
s = "Python "  
print(s * 3) # Natija: Python Python Python
```

Bo'lish (/). Bu operator bo'lish amalini bajaradi va haqiqiy son (float) qaytaradi.

```
a = 10  
b = 4  
c = a / b  
print(c) # Natija: 2.5
```

Butun bo'lish (//). Bu operator bo'lish natijasining butun qismini qaytaradi.

```
a = 10  
b = 4  
c = a // b  
print(c) # Natija: 2
```

Agar manfiy sonlar bilan ishlatilsa:

```
x = -10
```

```
y = 3
print(x // y) # Natija: -4
```

(Manfiy sonlarda katta butun songa qarab yaxlitlanadi!)

Qoldiq olish (%). Bu operator bo'lishning qoldiq qismini qaytaradi.

```
a = 10
b = 4
c = a % b
print(c) # Natija: 2
```

Amaliyotda toq va juft sonlarni tekshirishda ishlatiladi:

```
n = 7
if n % 2 == 0:
    print("Juft son")
else:
    print("Toq son") # Natija: Toq son
```

Darajaga oshirish (**). Bu operator darajaga oshirish uchun ishlatiladi.

```
a = 3
b = 2
c = a ** b
print(c) # Natija: 9 (3^2)
```

Kvadrat ildiz olish uchun (1/2 darajasi) ishlatiladi:

```
x = 16
print(x ** 0.5) # Natija: 4.0
```

Oddiy belgilash operatori (=). Bu operator o'zgaruvchiga qiymat tayinlash uchun ishlatiladi.

```
x = 10 # x ga 10 qiymati berildi
y = 5  # y ga 5 qiymati berildi
print(x, y) # Natija: 10 5
```

Qo'shish va belgilash (+=). Bu operator o'zgaruvchiga berilgan qiymatga yangi qiymatni qo'shib, natijani saqlaydi.

```
x = 10
x += 5 # x = x + 5
print(x) # Natija: 15
```

Ayirish va belgilash (-=). Bu operator o'zgaruvchining qiymatidan

berilgan qiymatni ayirib, natijani o'zgaruvchiga saqlaydi.

```
x = 10
x -= 3 # x = x - 3
print(x) # Natija: 7
```

Ko'paytirish va belgilash (*=). Bu operator o'zgaruvchini berilgan qiymatga ko'paytirib, natijani o'zgaruvchiga saqlaydi.

```
x = 5
x *= 3 # x = x * 3
print(x) # Natija: 15
```

Bo'lish va belgilash (/=). Bu operator o'zgaruvchini berilgan qiymatga bo'lib, natijani saqlaydi (**haqiqiy son qaytaradi**).

```
x = 10
x /= 4 # x = x / 4
print(x) # Natija: 2.5
```

Butun bo'lish va belgilash (//=). Bu operator butun bo'lishni bajarib, natijani o'zgaruvchiga saqlaydi.

```
x = 10
x //= 3 # x = x // 3
print(x) # Natija: 3
```

Qoldiq olish va belgilash (%=). Bu operator bo'lish qoldig'ini hisoblab, natijani o'zgaruvchiga saqlaydi.

```
x = 10
x %= 3 # x = x % 3
print(x) # Natija: 1
```

Darajaga oshirish va belgilash (**=). Bu operator o'zgaruvchini berilgan qiymat darajasiga oshirib, natijani o'zgaruvchiga saqlaydi.

```
x = 4
x **= 3 # x = x ** 3
print(x) # Natija: 64 (4^3)
```

Python-dagi matematik funksiyalar va konstantalar

Python matematik amallar uchun math moduli bilan ishlaydi. Bu modul trigonometrik, logarifmik, ildiz chiqarish va boshqa matematik funksiyalarni o'z ichiga oladi.

Matematik konstantalar

Konstanta	Tavsifi	Qiymati
math.pi	Pi soni (π)	3.141592653589793
math.e	Natural logarifm asosi (e)	2.718281828459045
math.tau	Tau ($\tau = 2\pi$)	6.283185307179586
math.inf	Cheksizlik (∞)	Infinity
math.nan	NaN (Not a Number)	NaN

```
import math
print(math.pi) # 3.141592653589793
print(math.e) # 2.718281828459045
print(math.inf) # inf (cheksizlik)
```

Asosiy matematik funksiyalar

Funksiya	Tavsifi	Misol	Natija
math.sqrt(x)	Kvadrat ildiz	math.sqrt(25)	5.0
math.pow(x, y)	Darajaga oshirish	math.pow(2, 3)	8.0
math.exp(x)	e^x hisoblash	math.exp(2)	7.389
math.log(x)	Natural logarifm ($\ln x$)	math.log(10)	2.302
math.log(x, b)	Logarifm ($\log_b(x)$)	math.log(8, 2)	3.0
math.factorial(n)	Faktorial	math.factorial(5)	120
math.gcd(a, b)	Eng katta umumiy bo'luvchi (EKUB)	math.gcd(48, 18)	6

```
import math
print(math.sqrt(16)) # 4.0
print(math.pow(2, 3)) # 8.0
print(math.log(100, 10)) # 2.0
```

```
print(math.factorial(5)) # 120
print(math.gcd(24, 36)) # 12
```

Trigonometrik funksiyalar

Funksiya	Tavsifi	Misol	Natija
math.sin(x)	Sinus	math.sin(math.pi/2)	1.0
math.cos(x)	Kosinus	math.cos(0)	1.0
math.tan(x)	Tangens	math.tan(math.pi/4)	1.0
math.asin(x)	Arcsinus	math.asin(1)	$\pi/2$
math.acos(x)	Arccosinus	math.acos(1)	0.0
math.atan(x)	Arctangens	math.atan(1)	$\pi/4$
math.degrees(x)	Radiandan darajaga o'tish	math.degrees(math.pi)	180.0
math.radians(x)	Darajadan radianga o'tish	math.radians(180)	π

```
import math
print(math.sin(math.pi/2)) # 1.0
print(math.cos(0)) # 1.0
print(math.tan(math.pi/4)) # 1.0
print(math.degrees(math.pi)) # 180.0
print(math.radians(90)) # 1.5708
```

Sonlar bilan ishlash

Funksiya	Tavsifi	Misol	Natija
math.ceil(x)	Yuqoriga yaxlitlash	math.ceil(4.2)	5
math.floor(x)	Pastga yaxlitlash	math.floor(4.7)	4
math.trunc(x)	Kasr qismini olib tashlash	math.trunc(4.9)	4
round(x, n)	n xonagacha yaxlitlash	round(3.14159, 2)	3.14

```
import math
print(math.ceil(4.2)) # 5
print(math.floor(4.7)) # 4
print(math.trunc(4.9)) # 4
print(round(3.14159, 2)) # 3.14
```


Tasodifiy sonlar bilan ishlash (random moduli). Python tasodifiy sonlar yaratish uchun random modulidan foydalanadi.

Funksiya	Tavsifi	Misol
random.random()	0 dan 1 gacha tasodifiy son	random.random()
random.randint(a, b)	a dan b gacha butun son	random.randint(1, 10)
random.uniform(a, b)	a dan b gacha haqiqiy son	random.uniform(1, 5)
random.choice(seq)	Ro'yxatdan tasodifiy element olish	random.choice([1, 2, 3])

```
import random
print(random.random())    # 0.2345 (har safar har xil)
print(random.randint(1, 10)) # 7 (tasodifiy)
print(random.uniform(1, 5)) # 3.456 (tasodifiy)
print(random.choice([1, 2, 3, 4])) # 2 (tasodifiy)
```

NAZORAT SAVOLLARI

- 1) Fizik jarayonlarni modellashtirish qanday maqsadlarda qo'llaniladi?
- 2) Python dasturlash tilining fizik jarayonlarni modellashtirishda afzalliklari nimalardan iborat?
- 3) Algoritm tuzishning bosqichlari qanaqa ekan?
- 4) Fizik modelning ta'rifi qanaqa?
- 5) Fizika va astronomiya fanlarining rivojlanishida modellashtirishning roli qanaqa?
- 6) Blok-sxmalarning asosiy elementlarini aytib bering
- 7) Matematik va trigonometric funksiyaar bilan ishlash uchun kutubxonalarning qaysi biriga murojlat qiish kerak?
- 8) Python kodlari qaysi kengaytma bilan saqlanadi?
- 9) Kodni bajarish natijasi qanday bo'ladi?
 $a = 5; b = -3; a = 8;$
 $c = a + b$
`print(c)`
- 10) `print(2 + 3 * 4)` kodi qanday natijani beradi?

2-MAVZU: Tarmoqlanuvchi algoritmlar, if va else shart operatrlari bilan ishlash misollari.

Reja:

1. Tarmoqlanuvchi algoritmlar, if va else shart operatrlari bilan ishlash misollari.
2. Chala tarmoqli va to'liq tarmoqli algoritmlar.
3. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash.
4. Mantiqiy operatsiyalar.
5. Sikllar bilan ishlash: sikllar va siklik operatorlar.
6. Sikllar yordamida summa va ko'paytmalarni topish.
7. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.

Tarmoqlanuvchi algoritmlar

Tushunarliki, tabiiy masalarni yechganda, sof chiziqli algoritmlar yetarli bo'lmaydi. Masalalarning ko'pchiligini yechganda tanlash, yoxud biron bir shartni tekshirish masalasi paydo bo'ladi. Masalan, oddiy kvadrat tenglamasini yechganimizda diskriminantning manfiy yoki musbat ekanligiga qarab, yechimlar haqida u yoki bu xulosaga kelamiz. Yana bitta misol – do'kondan biron narsani sotib olmoqchi bo'lsak, avval mablag'imiz yetadimi degan savolga javob qidiramiz. Ya'ni xatti-harakatlarimiz doimo paydo bo'ladigan shartlarga bog'liq, xulosa qilish va yechimga kelish esa imkoni bo'lgan yo'llar – tarmoqlarning birortasini tanlash masalasiga taqaladi. Shunda bizga tarmoqlanuvchi algoritmlar yordam beradi (va nafaqat dasturlashda).

Hozir biz munosabatlar operatorlaridan boshlaymiz (5- jadval). Munosabatlar operatorlari ikkita kattalik, o'zgaruvchi, vektor yoki matritsani bir-biri bilan solishtirishda xizmat qiladi.

5- jadval. Munosabatlar operatorlari (a = 3; b = 4; c = 3;) # belgi bilan javobi nima bo'lishi kerakligi ko'rsatilgan			
Belgi	Vazifasi	Sintaksisi va misoli	
==	Tenglik. Ikkita kattalikning tengligini tekshiradi. Yodda tuting, bu ikkita kattalikni tenglashtirmaydi va o'zgartirmaydi, faqat solishtiradi.	a = 3; b = 4; c = 3; w = (a==c) # True	
!=	Teng emaslik. Ikkita kattalikning teng emasligi tekshiradi.	q = (a!=c) # False	q = (a!=b) # True
>	Katta. Chapda a son c sonidan kattami,	n = (a>c)	n = (b>a)

	degan savolga yolg'on (0) degan javob berildi. O'ng tomonda b son a son dan kattami, degan savolga to'g'ri (1) degan javob berildi.	# False	# True
>=	Katta yoki teng. Oldingi misoldek a son c va b son dan katta yoki tengmi, degan savol qo'yilyapti va mos javoblar olinyapti.	$q = (a \geq c)$ # True	$q = (a \geq b)$ # False
<	Kichik. Chapda a son c son dan kichikmi, degan savolga yolg'on (0) degan javob berildi. O'ng tomonda a son b son dan kichikmi, degan savolga to'g'ri (1) degan javob berildi.	$n = (a < c)$ # False	$n = (a < b)$ # True
<=	Kichik yoki teng. Oldingi misoldek a son c va b son dan kichik yoki tengmi, degan savol qo'yilyapti va mos javoblar olinyapti.	$n = (a \leq c)$ # True	$n = (a \leq b)$ # True

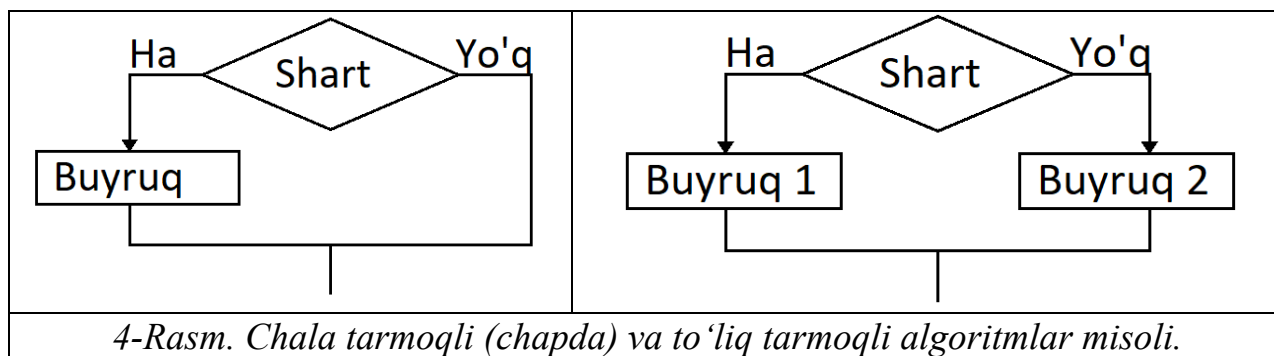
Misollarda ko'rganimizdek, munosabatlar operatorlari kattaliklarning yoki bir xil o'chamdagi vektor va matritsalarini elementlar bo'ylab solishtiradi va mantiqiy ifodaning javobi ikkitadan bittasi bo'la oladi: 1 (ha, haqiqat) yoki 0 (yo'q, yolg'on).

Kattaliklar kompleks sonli bo'lsa, == va ~= operatorlar haqiqiy va mavhum qismlarini alohida solishtirishadi. Faqat bunda alohida import cmath kutubxonasiga murojlat qilishni unutmang.

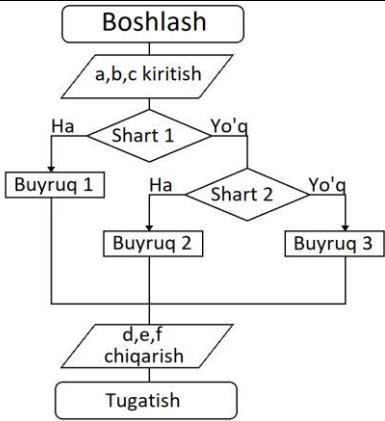
$a = 5 + 6i$; $b = -9 + 3i$; $c = 5 + 6i$; $d = 5 + 8i$; $e = -9 + 5i$;

$w = (a == b)$ # False	$w = (a == c)$ # True	$w = (a != d)$ # True	$w = (b == d)$ # False
---------------------------	--------------------------	--------------------------	---------------------------

Tarmoqlanuvchi algoritmlar, odatda, if va ba'zi hollarda switch nomli shartli operatorlar yordamida amalga oshiriladi. Shart yoki shartlarga qarab buyruqlarning u yoki bu ketma-ketligi oshirilishi kerak bo'lganda tarmoqlanish yuzaga keladi. Tarmoqlarning soniga qarab algoritmlar: a) chala tarmoqli (bitta tarmoq); b) to'liq tarmoqli (ikkita tarmoq); c) ko'p tarmoqli (ikkitadan ko'p tarmoq) bo'ladi.



6-Jadval. MATLABda tarmoqlanuvchi algortimlarni tashkil qilish misollari		
if <shart> <buyruqlar> end		
Sintatksis	Blok-sxema	Misol
if <shart> <buyruqlar> end		# Ikkita a va b sonni solishtirib, # a ning b ga nisbatan holatini # aniqlang a = 1; b = 3; if (a > b): print('a katta b dan'); if (a < b): print('a kichik b dan'); if (a == b): print('a teng b');
if <shart> <buyruq1> else <buyruq2> end		
if <shart> <buyruqlar 1> else <buyruqlar 2> end		% Ikkita a va b sonni solishtirib, ular teng yoki teng emasligini aniqlang a = 1; b = 3; if (a == b): print('a teng b ga'); else: print('a teng mas b ga');

7-Jadval. MATLABda ko'p tarmoqli algortimlarning misollari		
if <shart> <buyruq1> elseif <buyruq2> else <buyruq3> end		
Sintatksis	Blok-sxema	Misol
<pre> if <shart 1> <buyruqlar 1> elseif <shart 2> <buyruqlar 2> else <buyruqlar 3> End </pre>	 <pre> graph TD Boshlash([Boshlash]) --> Kiritish[/a,b,c kiritish/] Kiritish --> Shart1{Shart 1} Shart1 -- Ha --> Buyruq1[Buyruq 1] Shart1 -- Yo'q --> Shart2{Shart 2} Shart2 -- Ha --> Buyruq2[Buyruq 2] Shart2 -- Yo'q --> Buyruq3[Buyruq 3] Buyruq1 --> Chiqarish[/d,e,f chiqarish/] Buyruq2 --> Chiqarish Buyruq3 --> Chiqarish Chiqarish --> Tugatish([Tugatish]) </pre>	<pre> % Ikkita a va b % sonni solishtiring a = 2; b = 4; if a > b: print('a katta b dan'); else: if (a < b): print('a kichik b dan'); else: print('a teng b'); </pre>
if <shart> <buyruq1> else <buyruq2> end		

4-rasmda tarmoqlanuvchi algortimlarning misollari ko'rsatilgan. Shart sifatida ma'lum bir mantiqiy ifoda ko'rsatiladi. Agar shart haqiqat bo'lsa, unda Buyruq bajariladi, aks holda dasturning keyingi operatoriga o'tiladi (chapdagi rasm). Agar shart haqiqat bo'lsa, u holda Buyruq-1, aks holda Buyruq-2 bajariladi (o'ngdagi rasm). MATLABda shartli operatorlarning sintaksisi ko'rinishi 6-Jadvalda ko'rsatilgandek bo'ladi.

Tarmoqlanuvchi algortimlarda ikkitadan ko'p tarmoqni shakllantirish uchun qo'shimcha else, elseif, switch..case operatoridan foydalanamiz. Dasturni tuzishda va o'qishda qulay bo'lishi uchun if ... end juftligini matnda chap tomondan tekislaymiz (7-jadval).

4-Misol. Uchta a, b, c son berilgan. Ular orasida eng katasini toping. Bu masalaga dasturni bir nechta variantda tuzsa bo'ladi. m deb ular orasida eng kattasini belgilaymiz. 1-yondoshishda avval a va b solishtiramiz; ularning kattasini topib, m deb belgilaymiz; keyin m va c sonlarni solishtirib, ular orasidagi kattasini topamiz. Ikkinchi yondashishda eng kattasi deb m = a deymiz. Keyin uni boshqa ikkita son bilan solishtiramiz.

<pre> # Arifmetik operatsiyalar a = 1.2; b = 6.3; c = -9.2; if (a > b): m = a else: m = b; if (c > m): m = c </pre>	<pre> # Arifmetik operatsiyalar m = a; if (c > m): m = c; if (b > m): m = b; </pre>
<pre> print('%4.4f, %4.4f va %4.4f sonlar orasida eng kattasi %4.4f ekan' % (a,b,c,m)); </pre>	

5-Misol. Biron bir tengsizlikning yechimi $(-3; \infty)$ intervalida bo'lsin. Dasturingiz tasodifiy tanlangan z soni shu oraliqda ekanmi yoki yo'qmi, aniqlab bersin.

Arifmetik operatsiyalar

```
z = float(input('Ixtiyoriy sonni kiriting = '));
```

```
if (z > -3):
```

```
    print('\n Z = %4.4f son (-3; inf) oralig'ida joylashgan \n' % z);
```

```
else:
```

```
    print('\n Z = %4.4f son (-3; inf) oralig'ida joylashmagan \n' % z);
```

Mantiqiy operatsiyalar

Dasturlashtiriladigan masala va vazifalar kamdan-kam hollarda oddiy. Ko'pgina dasturlarda ma'lum bir talabga erishilgunga qadar bir nechta buyruqlar takrorlanadigan bir qator sikllar va bir nechta harakatlarning qaysi biri bajarilishi kerakligini aniqlaydigan turli xil tekshiruvlardan o'tish kerak bo'ladi. Eng sodda misol – biron bir tengsizlikning yechimi aniq belgilangan diapazonda yoki bir nechta diapazonlarda yotgan bo'lishi mumkin. Shunda sinov nuqtaning tegishlilikini tekshirish uchun bir nechta bosqichlardan o'tish kerak bo'ladi.

Mantiqiy iboralar ikkita haqiqat ma'nosidan birini olishi mumkin — "haqiqat" yoki "yolg'on". Mantqiy operatsiyalar oddiylardan murakkab mantiqiy ifodalarni ta'riflashga va yechishga xizmat qiladi. O'z navbatida, mantiqiy iboralar odatda dasturni bajarish ketma-ketligini boshqarish shartlari sifatida ishlatiladi.

Matematikada eng sodda amallar – qo'shish va ko'paytirish. Shunga o'xshash operatsiyalar dasturlashda ham ishlatiladi va mantiqiy operatsiyalar deyiladi. Ammo bu yerda qo'shish va ayirish amallari sonlar ustida emas, balki operandlar ustida bajariladi.

Operandlar – har qanaqa bayonot bo'lib, ularning qiymatlari haqiqat (1) yolg'on (0) bo'lishi mumkin. Ushbu qiymatlar ma'lum operatsiyalar qonunlariga bo'ysunib, dasturlashdagi voqealar mantiqini belgilaydi. Qanday operatsiyalar mavjudligini va ular nima uchun kerak ekanligini ko'rib chiqamiz.

Mantiqiy operatsiya – bu operatsiyalar ustida amal bo'lib, yangi mantiqiy kattalikni qaytaradigan operatsiya. Demak, agarda biz ikkita

bayonotga asoslanib, yangi bayonotni mantiqiy hosil qilsak, mantiqiy operatsiya'ni bajargan bo'lamiz.

Agarda biron bir gap haqida "haqiqat" yoki "yolg'on" deb aytish mumkin bo'lsa, bu gapni bayonot deb tushunamiz. Bundan tashqari, har qanaqa bayonot yoki "to'g'ri" yoki "yolg'on" bo'lishi kerak. Hech qanaqa bayonot bir vaqtning o'zida ham "to'g'ri" ham "yolg'on" bo'la olmaydi. Bayonotlarning misoli:

1) Toshkent – O'zbekiston poytaxti;

2) 27 bu oddiy son;

3) $3 > 5$.

1)- va 3) bayonotlar "to'g'ri", 2)- "yolg'on", chunki $27 = 3 * 3 * 3$. Quyidagi gap va ifodalar: a) darsni o'zlashtirish; b) $2 * x > 8$; c) $a * x^2 + b * x + c = 0$; d) Soat necha? Bayonot bo'la olmadi, chunki bular haqida "to'g'ri" yoki "yolg'on" deb ayta olmaymiz.

Bayonotlar yordamida obyektlarning xususiyatlari, ular orasidagi bog'lanishlar aniqlanadi. Agarda bayonot bunday bog'lanishni to'g'ri aks ettirsa, u "to'g'ri", aks holda "yolg'on" bo'ladi. Suv harorati 20 gradus, Yerdan Quyoshgacha masofa 150 mln km, $3 > 5$ kabi bayonotlar aniq obyektning xususiyatini bildiradi (havo, Koinot, sonlar). Ularning har biri "to'g'ri" yoki "yolg'on" bo'lishi mumkin.

Mantiqiy operatsiyalarda bayonotlarning mazmuni qaralmaydi, buning o'rniga ularning "to'g'ri" yoki "yolg'on" ekanligi muhim. Shuning uchun bayonotning qiymati, yuqorida aytganimizdek, 1 (to'g'ri) yoki 0 (yolg'on) teng bo'ladigan oddiy o'zgaruvchan deb qaraladi. Bu o'zgaruvchilar kombinatsiyalaridan tashkil topgan murakkab bayonotlar mantiqiy funksiyalar bo'ladi. Tabiiyki, murakkab mantiqiy funksiyalarning to'g'ri yoki yolg'onligi ushbu murakkab bayonotlarni tashkil qilgan oddiy bayonotlarning to'g'ri yoki yolg'onligiga bog'liq.

Mantiqiy o'zgaruvchilarning haqiqiyligi va yo'g'onligini ifodalash uchun turli belgilashlar mavjud:

Haqiqat	H	True	T	1
Yolg'on	Y	False	F	0

Mantiqiy funksiyalardagi oddiy bayonotlar bir-biri bilan asosiy uchta operatsiyalar bilan bog'lanadi

AND (VA, &) – mantiqiy ko'paytirish (konyunksiya);

OR (YOKI, |) – mantiqiy qo'shish (dizyunksiya);

NOT (EMAS, ~) – mantiqiy rad etish (inversiya).

Ularni alohida misollar bilan ko'rib chiqamiz.

1) mantiqiy ko‘paytirish – konyuksiya lotin tilidagi conjunctio – bog‘lanish so‘zidan olingan. A = “Tashqarida quyosh charaqlayapti”, B = “Tashqarida havo ochiq” degan ikkita oddiy bayonot “AND” ulagichi bilan bir biriga bog‘lanadi va natijada “Tashqarida quyosh charaqlayapti va havo ochiq” degan murakkab bayonotga kelamiz. MATLABda AND operatsiyasi & belgi yordamida amalga oshiriladi.

Konyuksiyadagi haqiqiylik jadvali bunday tasvirlanadi

Murakkab Bayonat	A	B	A&B
Tashqarida Quyosh yoritmayapti va havo bulutli	0	0	0
Tashqarida Quyosh yoritmayapti va havo ochiq	0	1	0
Tashqarida Quyosh charaqlayapti va havo bulutli	1	0	0
Tashqarida quyosh charaqlayapti va havo ochiq	1	1	1

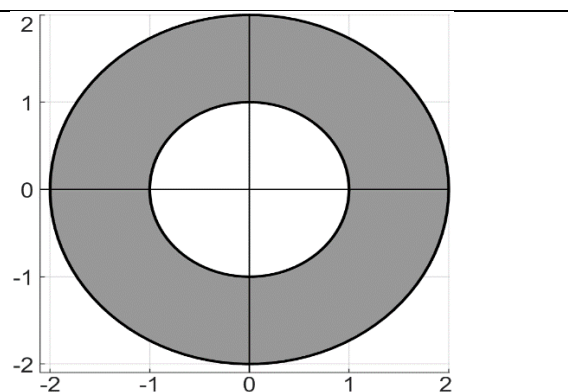
Konyuksiyada tashkil qiluvchi oddiy bayonotlarning ikkalasi ham “to‘g‘ri” bo‘lsagina, mantiqiy funksiya ham ”to‘g‘ri” bo‘ladi, aks hollarda u doimo “yolg‘on” bo‘lib qolaveradi.

6-Misol. Tengsizlikning yechimi $x \in (3.5; 8.2)$ intervalda bo‘lsin. Ixtiyoriy sinov nuqtasi shu intervalga kirishi yoki yo‘qligini aniqlaydigan dasturni tuzing.

```
x = float(input('sinov nuqtaning kattaligini kiriting = '));
a = 3.5; b = 8.2; # intervalning chegaralari belgilandi
if ((x > a) & (x < b)):
    print('\n X = %2.1f son (%2.1f; %2.1f) intervalga kiradi \n' % (x,a,b) );
else:
    print('\n X = %2.1f son (%2.1f; %2.1f) intervalga kirmaydi \n' % (x,a,b) );
```

Mantiqiy shartimiz endi & ulagich bilan bog‘langan ikkita qismdan iborat ($x > a$) va ($x < b$). Sinov nuqtamiz shu intervalda bo‘lishi uchun u bir vaqtning o‘zida 3.5 dan katta va 8.2 dan kichik bo‘lishi shart. Shundagina murakkab bayonot “to‘g‘ri” bo‘ladi.

7-Misol. Sinov nuqta rasmda bo‘yalgan sohaga tegishliligini aniqlang



```
import random as rnd
import math as mt
x = rnd.uniform(-2,2); # Sinov buqtaning X koordinatasi tasodifiy
tanlanadi
y = rnd.uniform(-2,2); # Sinov buqtaning Y koordinatasi tasodifiy
tanlanadi
R1 = 1; R2 = 2; # Sohaning ichki va tashqi radiuslari
r = mt.sqrt(x*x + y*y); #Sinov nuqtaning aylanalar markazigacha masofa
if (r > R1) & (r < R2):
    print('\n Nuqta = (%2.1f; %2.1f) halqaning ichida \n' % (x,y) );
else:
```

```
    print('\n Nuqta = (%2.1f; %2.1f) halqadan tashqarida \n' % (x,y) );
```

Sinov naqtamiz bo‘yalgan sohada bo‘lishi uchun uning markazgacha bo‘lgan masofasi birinchi radiusdan ($R1 = 1$) katta va ikkinchi radiusdan ($R2 = 2$) kichik bo‘lishi kerak.

2) mantiqiy qo‘shish – dizyunksiya lotin tilidagi disjunctio – ajratish so‘zidan olingan. $A =$ “Tashqarida quyosh charaqlayapti”, $B =$ “Tashqarida havo bulutli” degan ikkita oddiy bayonot “OR” ulagichi bilan bir-biriga bog‘lanadi va natijada “Tashqarida quyosh charaqlayapti yoki havo bulutli” degan murakkab bayonotga kelimiz. MATLABda OR operatsiyasi | belgi yordamida amalga oshiriladi.

Dizyunksiyadagi haqiqiylik jadvali bunday tasvirlanadi

Murakkab bayonot	A	B	A B
Tashqarida quyosh yoritmayapti yoki havo ochiq	0	0	0
Tashqarida quyosh yoritmayapti yoki havo bulutli	0	1	1
Tashqarida quyosh charaqlayapti yoki havo ochiq	1	0	1

Tashqarida quyosh charaqlayapti yoki havo bulutli	1	1	1
---	---	---	---

Konyuksiyada tashkil qiluvchi oddiy bayonotlarning ikkalasi ham “yolg‘on” bo‘lsagina, mantiqiy funksiya ham ” yolg‘on” bo‘ladi, aks hollarda u doimo “to‘g‘ri” bo‘lib qolaveradi.

8-Misol. Tengsizlikning yechimi $x \in (-\infty; -2.6] \cup (3.5; \infty)$ intervalda bo‘lsin. Ixtiyoriy sinov nuqtasi shu intervalga kirishi yoki yo‘qligini aniqlaydigan dasturni tuzing.

```
import random as rnd
```

```
x = rnd.uniform(-8,8); # Sinov buqtaning X koordinatasi tasodifiy tanlanadi
```

```
a = -2.6; b = 3.5; # intervallarning ichki chegaralari belgilandi
```

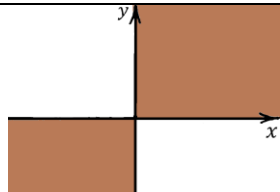
```
if ( (x <= a) or (x > b) ):
```

```
    print('\n X = %2.1f son (-Inf;%2.1f) U (%2.1f, +Inf) intervalga kiradi\n' % (x,a,b) );
```

```
else:
```

```
    print('\n X = %2.1f son (-Inf;%2.1f) U (%2.1f, +Inf) intervalga kirmaydi\n' % (x,a,b) );
```

Mantiqiy shartimiz endi | ulagich bian bog‘langan ikkita qismdan iborat ($x < a$) va ($x > b$). Sinov nuqtamiz shu intervalda bo‘lishi uchun u bir vaqtning o‘zida -2.6 dan kichik yoki 3.5 dan katta-kichik bo‘lishi mumkin. Shunda murakkab bayonot “to‘g‘ri” bo‘ladi.

9-Misol. Sinov nuqta rasmda bo‘yalgan soha (ya’ni 1- va 3-choraklar) ga tegishliligini aniqlang.	
--	--

```
import random as rnd
```

```
x = rnd.uniform(-2,2); # Sinov buqtaning X koordinatasi tasodifiy tanlanadi
```

```
y = rnd.uniform(-2,2); # Sinov buqtaning Y koordinatasi tasodifiy tanlanadi
```

if ((x>0 and y>0) or (x<0 and y<0)):

```
print('\n Nuqta = (%2.1f; %2.1f) halqaning ichida \n' % (x,y) )
```

else:

```
print('\n Nuqta = (%2.1f; %2.1f) halqaning tashqarisida \n' % (x,y) )
```

Sinov naqtamiz bo‘yalgan sohaga bo‘lishi uchun uning ikkala koordinatasi bir vaqtning o‘zida musbat yoki manfiy bo‘lishi kerak.

3) mantiqiy rad etish – inversiya lotin tilidagi inversion– almashtirish, aylantirish so‘zidan olingan. Oddiy tilda EMAS so‘ziga mos keladi. Masalan, A = 10 – juft son degan bayonot “haqiqat” qiymatiga ega. Buni rad etish B = 10 – toq son bo‘lib, qiymati “yolg‘on” ga teng. Va aksincha, A = 10 – toq son degan bayonot “yolg‘on” va uni rad etgan B = 10 – juft son bayonoti “haqiqat” ga aylanadi. MATLABda EMAS operatsiyasi ~ belgi yordamida amalga oshiriladi.

Inversiyadagi haqiqiylik jadvali bunday tasvirlanadi

A	B = ~ A
1	0
0	1

Agar asl ifoda “to‘g‘ri” bo‘lsa, natija “yolg‘on” bo‘ladi, va aksincha.

Shu uchta mantiqiy operatsiyadan tashqari, mantiqiy ta‘qib qilish (implikatsiya), mantiqiy tenglik (ekvivalentlik) kabi boshqa operatsiyalar ham mavjud. Ular haqida batafsilroq bilmoqchi bo‘lsangiz, adabiyotlarga murojaat qilishiz mumkin. Ushbu qo‘llanmadagi ma‘lumotlarni o‘zlashtirish uchun aytilgan 3 ta operatsiya yetarli deb o‘ylaymiz. Rad etish operatsiyalariga misollarni keyingi paragrafdagi misollar bilan birga beriladi.

Sikllar va siklik operatorlar

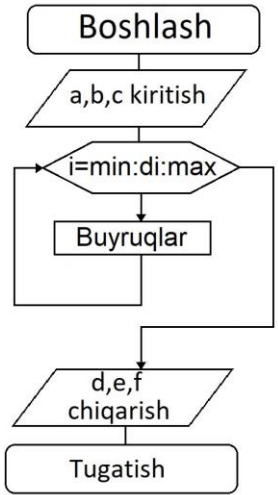
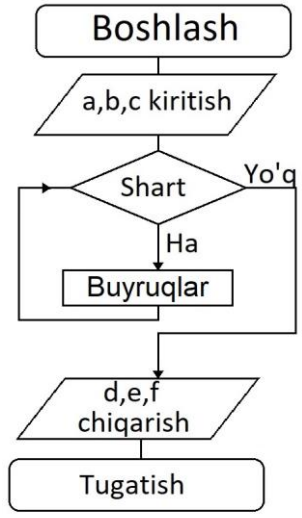
Masalani yechish davomida dasturdagi ma‘lum bir buyruqni yoki buyruqlarni (blokini) bir necha marta bajarishingiz kerak bo‘lgan vaziyat yuzaga kelishi mumkin. Odatda, operatorlar ketma-ket bajariladi. Birinchidan, funksiyadagi birinchi operator bajariladi, undan keyin ikkinchisi va boshqalar. Agarda ularning har birini alohida yozadigan bo‘lsak, juda ko‘p joy talab etiladi. Uning ustiga aynan nechi marta takrorlash keraklini oldindin bilmasligimiz ham mumkin.

Dasturlash tillari yanada murakkab ijro yo‘llarini yaratishga imkon beradigan turli xil boshqaruv tuzilmalarini taqdim etadi. Siklik operatori

bizga buyruq yoki yoki buyruqlar ketma-ketligini bir necha marta bajarishga imkon beradi. 5-rasmda ko'pgina dasturlash tillari uchun siklik operatorining umumiy blok-sxemalari ko'rsatilgan.

Python sikl talablarini bajarish uchun turli xil shakllarni taqdim etadi, shu jumladan for... va while... sikllari. Aynan shular eng ko'p ishlatiladigan operatorlar. Takrorlash soni oldindan ma'lum bo'lganda for sikliklik operatori ishlatiladi. Ma'lum bir shart haqiqiy bo'lgunicha biron bir buyruqlarni bajarish talab etiladigan bo'lsa, while operatorini ishlatamiz. Bu holda aynan nechi marta takrorlash keraklini ma'lum bo'lmasligi ham mumkin. for va while sikllarning shartidan keim ikki nuqta belgisi qo'yiladi : bu sikllarning boshlanishini ko'rsatadi, undan keyin sikl tanasi keladi va uning yozishda kamida 1 probel bilan o'ng tomomda bo'ladi.

Aytganimizdek, for siklik operatori buyruq yoki buyruqlar majmuasini oldindan belgilangan marta bajaradi. for kalit so'zidan keyin indeks – sikl sanagichi va uning qiymatlari diapazoni ko'rsatiladi. Musbat indekslar uchun uning kattaligi maksimal qiymatidan oshganda siklning bajarilishi to'xtatiladi, manfiylarda esa – teskarisi, indeks ko'rsatilgan chegaradan kamayib ketguncha sikl takrorlanaveradi.

5-Rasm. for...end (chapda) va while...end (o'ngda) ko'rinishdagi sikllar uchun blok-sxemalar ko'rsatilgan.	
 <pre> graph TD Start([Boshlash]) --> Input[/a,b,c kiritish/] Input --> Decision{i=min:di:max} Decision --> Statements[Buyruqlar] Statements --> Decision Decision --> Output[/d,e,f chiqarish/] Output --> End([Tugatish]) </pre>	 <pre> graph TD Start([Boshlash]) --> Input[/a,b,c kiritish/] Input --> Decision{Shart} Decision -- Ha --> Statements[Buyruqlar] Statements --> Decision Decision -- Yo'q --> Output[/d,e,f chiqarish/] Output --> End([Tugatish]) </pre>
<pre> for i in range(1,10): buyruq_1; buyruq_2; # 10 siklda ishtirok etmaydi </pre>	<pre> i = 0; while(i < 10): buyruq_1; buyruq_2; i=i+1 </pre>

9-Misol. Ixtiyoriy butun n uchun uning $n!$ faktorialini toping

```
# Faktorilani hisoblaydigan sikl
# n! topish kerak, bu yerda n – butun musbar son
n = int(input('Faktorial asosini kiriting n = '));
res = 1;
for i in range(1, n+1):
    res = res * i;
print('Natijada %d! = %d \n' % (n,res) )
```

Agarda diapazondagi qadam ko'rsatilmasa, u avtomatik ravishda $di = 1$ deb hisoblanadi.

10-Misol. $n = 0$ dan ixtiyoriy chegaragacha bo'lgan sonlarning yig'indisini toping.

```
# n=0 dan ixtiyoriy chegaragacha
# sonlarning yig'indisini toping: n = SUM(0,INF)
n = int(input('Sonlarning yuqori chegarasini kiriting n = '))
res = 0;
for i in range (0, n+1):
    res = res + i;
print('Natija = %d ' % (res) )
```

Siklik operator while shunday ishlaydiki, operator tanasidagi buyruqlar ma'lum shartlar to'g'ri bo'lsagina, bajarila boshlanadi va takrorlanadi. Ko'rsatilgan shart yolg'onga aylanganida, bu operator o'z ishini to'xtatadi. Shunda while siklik operatori takrorlanishlar soni oldindan ma'lum bo'lmagan hollarda ishlatiladi.

Bu sikl ishlatilganda oldin qo'yilgan shart tekshiriladi. Agarda u yolg'on bo'lsa, unda sikl boshlanmasdan tugaydi va end kalit so'zidan keyin kelayotgan buyruq bajariladi

11-Misol. Ikkita sonning eng katta umumiy bo'luvchisini topadigan dasturni yozing. Bu masalani yechish uchun Evklid usullariga murojaat qilamiz

% a va b sonlarining eng katta umumiy bo'luvchisini hisoblash	% a va b sonlarining eng katta umumiy bo'luvchisini hisoblash
---	---

<pre> a = 6; b = 42; while (a != b): if (a > b): a = a - b else: b = b - a #Natijani NOD o'zgaruvchisi bilan saqlaymiz NOD = a ; print(NOD); </pre>	<pre> a = 30; b = 20; while (a != 0) and (b != 0): if (a > b): a = (a%b); else: b = (b%a); # Natijani NOD o'zgaruvchisi bilan saqlaymiz NOD = a + b; print(NOD); </pre>
--	---

12-Misol. Teylor qatoridan foydalanib $1/\sqrt{1+x}$ funksiya qiymatining $(-1 < x \leq 1)$ oraliqdagi qiymatini toping va o'rnatilgan funksiyalar beradigan qiymati bilan solishtiring. Hisoblashda aniqlikni $|S_n - S_{n-1}| = \varepsilon = 0.001$ deb oling. Berilgan funksiyaning Teylor qatori

$$\frac{1}{\sqrt{1+x}} = \sum_{n=0}^{\infty} \frac{(-1)^n (2n)!}{2^{2n} (n!)^2} (x)^n$$

```

import math as mt
x = float(input('Argument qiymatini kiriting x = '))
# Masalada berilgan aniqlik mezon
epsilon = 0.0001;
# summadagi joriy va oldingi qiymatlari uchun o'zgaruvchilar
ss = 0; ss_1 = 0;
# shu summalarning ayirmasi uchun o'zgaruvchi kiritildi, siklning sharti
# bajarilishi uchun 1-yaqinlashda uning qiymatini sun'iy ravishda
ko'tarildi
diff_sum = 1000;
# Formulaga binoan birinchi had n = 0 dan boshlanadi
n = 0;
while (diff_sum > epsilon):
    # Murakkab funksiya bo'lgani uchun surat v maxraj qismlarini alohida
    # hisoblashni tavsiya qilamiz
    surat = mt.pow(-1,n) * mt.factorial(2*n) * mt.pow(x,n);
    maxraj = mt.pow(2,2*n) * mt.pow(mt.factorial(n),2);
    ss = ss + surat / maxraj;
    diff_sum = abs(ss - ss_1);
    ss_1 = ss;
    n = n + 1;

```



```

javob = ss;
print('Korsatilgan summaning x=%3.1f nuqtadagi qiymati %3.4f ga teng
\n' % (x,javob) )
# O'rnatilgan funksiya orqali hisoblab ko'ramiz va natijalarni solishtirib
ko'ramiz
javob2 = 1/(mt.sqrt(1+x));
print('Masaladagi funksiyaning x=%3.1f nuqtadagi qiymati %3.4f ga teng
\n' % (x,javob2) )

```

```

abs_farq = abs(javob - javob2);
print('Ikkita natija orasidagi absolyut farqi = %3.12f ga teng \n' %
(abs_farq) )

```

```

nisb_farq = abs(javob - javob2)/abs(javob);
print('Ikkita natija orasidagi nisbiy farqi = %3.12f ga teng \n' %
(nisb_farq))

```

12-Misoldagi kabi masalalarni yechganda nimani topish kerakligiga e'tibor qiling: yig'indinimi yoki ko'paytmanimi? Yig'indi bo'lganda, birinchi had $ss = 0$, ko'paytma bo'lganda $pp = 1$ bo'lishini inobatga oling. Aks holda, noto'g'ri javob qaytariladi.

```

for i in range(1,11):
    if (i == 3 ):
        continue
    print(i);
    if (i == 7):
        break

```

for va while sikllarda ularning bajarilishiga ta'sir etadigan operatorlar ham ishlatiladi. Biron bir qo'shimcha shart to'g'ri bo'lib chiqsa, siklni to'xtatish kerak bo'lsa, break operatoridan foydalanamiz. Sikldagi joriy qadamni rad etib, keying qadamga o'tish talab etilsa, continue operatoridan foydalanamiz. pause operatorini ishlatsak, aytilgan shart bajarilganda sikl vaqtincha ishini to'xtatadi, ammo klaviaturadagi ixtiyoriy klavishani bosganimizda, siklning ishi davom etaveradi. Ining ustiga pause (t), deb yozsak, to'xtab turish vaqtini boshqargan bo'lamiz, chunki bu yerda t – sekund birligida berilgan vaqt. Masalan, 1 dan to 10 gacha sonlarni ekranga chiqarilsin, ammo son 3 teng bo'lganda haq o'tkazib yuborilsin, 7 teng bo'lganda umuman to'xtatilsin, degan qo'shimcha shartlar kiritilgan masalaga dastur misol qilib ko'rsatilgan.

Oddiy sikllardan tashqari, ichki sikllar ham bo'lishi mumkin. Buni bitta sikl boshqa sikl ichida ishlayapti deb qarash kerak. Buni 11-misolda

koʻrdik, faqat u yerda sikl ichida oddiy mantiqiy tekshirish ketyapti. Toʻliq vaziyatni quyidagi masalani yechib, 13-misol qilib koʻrsatamiz. Oddiy gap bilan aytganda, bu yerda (a) summalarning koʻpaytmasini va (b) koʻpaytmalarning yigʻindisini topish kerak ekan.

13-Misol. Yigʻindining koʻpaytmasini va koʻpaytmaning yigʻindisini toping	
a	$P = \prod_{j=1}^k \frac{(j - j^2)k}{j - 4} \sum_{i=j}^{k+2} \frac{\sqrt[3]{ i - 5 }}{ i - 7 }$ <pre> import math as mt k = int(input('Qatorlarning yuqori chegarasini kiriting = ')) P = 1; # koʻpaytma boʻlgani uchun birinchi had 1 teng for j in range(1, k): # tashqi sikl boshlanishi S = 0; # Summa boʻlgani uchun birinchi had 0 teng for i in range(j, k+3): # Ichki sikl boshlanishi surat_1 = mt.pow(abs(i-5),1/3); maxraj_1 = abs(i-7); # Shu kabi shart bilan nolga aylanadigan maxrajlarni rad etamiz if (maxraj_1 == 0): continue; S = S + surat_1/ maxraj_1; # Koʻpaytma boʻlgani uchun suratni ham tekshirishimiz kerak if (S == 0): continue surat_2 = j - j^2; maxraj_2 = j - 4; if (surat_2 == 0 or maxraj_2 == 0): continue P = P * S * k * surat_2 / maxraj_2; print('Masalaning javobi = %3.4f\n\n' % (P)) </pre>
b	$W = \sum_{k=1}^n \frac{(-2)^{3k+1}}{4(k-1)x^{3k+1}} \prod_{m=1}^{k+2} \frac{m^3 - 8}{m - 4}$ <pre> import math as mt n = int(input('Qatorlarning yuqori chegarasini kiriting = ')) x = - 1.2; # Buni (-2;2) oraligdatanlasa boʻladi # Summa boʻlgani uchun birinchi had 0 teng S = 0; </pre>

```

for k in range(1 , n): # tashqi sikl boshlanishi
    # Ichki sikl boshlanishi
    P = 1; # Ko'paytma bo'lgani uchun birinchi had 1 teng
    for m in range(1 , k+3):
        surat_1 = mt.pow(m,3) - 8;
        maxraj_1 = m - 4;
        # Shu kabi shart bilan nolga aylanadigan maxraj va suratni rad etamiz
        if (surat_1 == 0 or maxraj_1 == 0):
            continue;
        P = P * surat_1/ maxraj_1;
        # Summa bo'lgani uchun maxrajning o'zini tekshirsak yetarli
        surat_2 = mt.pow(-2,3*k+1);
        maxraj_2 = 4 * (k-1) * mt.pow(x,3*k+1);
        if (maxraj_2 == 0):
            continue
        S = S + P * surat_2 / maxraj_2;
print('Masalaning javobi = %3.4f\n\n' % (S) )

```

NAZORAT SAVOLLARI

1. Qanday mantiqiy munosabatlar bor?
2. Tarmoqlanuvchi algoritmnining blok-sxamasini ko'rsating.
3. Tarmoqlanuvchi algoritmnining operatorlari qanaqa?
4. Mantiqiy operatsiya – bu qanaqa operatsiya?
5. Mantiqiy ko'paytirish (konyuksiya) operatorini qanday?
6. Mantiqiy qo'shish (dizyunksiya) operatorini qanday?
7. Kodni bajarish natijasi qanday bo'ladi?

b = 12

if a % 3 == 0:

print("Yes")

else:

print("No")

8. Sikl nechki marta bajariladi ?

for k in range(7,12):

9. Siklda continue operatori nima qiladi?

10. Siklda break operatori nima qiladi?

11. Mantiqiy masalaning yechimini ko'rsating

a = True; b = True

result = a and b

```
print(result)
```

12. Mantiqiy masalaning yechimini ko'rsating

```
a = False; b = False
```

```
result = a and b
```

```
print(result)
```

13. Mantiqiy masalaning yechimini ko'rsating

```
a = True; b = True
```

```
result = a or b
```

```
print(result)
```

14. Mantiqiy masalaning yechimini ko'rsating

```
a = False; b = True
```

```
result = a or b
```

```
print(result)
```

3-MAVZU: MASSIVLAR BILAN ISHLASH. GRAFIKLAR CHIZISH USULLARI (2 soat)

Reja:

1. Bir o'lchamli va ikki o'lchamli massivlarni shakillantirish
2. Elementlari bilan ishlash.
3. Tasodifiy jarayonlar xaqida tushuncha va tasodifiy sonlar bilan ishlash.
4. Grafiklar chizish usullari.
5. matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

Massivlar

Massiv nima?

Massiv - bu bir xil turdagi ma'lumotlarni (masalan, sonlar, matnlar, obyektlar) ketma-ket saqlash uchun ishlatiladigan ma'lumotlar strukturasi. Massivlarning har bir elementi indeks orqali murojaat qilinadi (odatda 0 dan boshlanadi).

Pythonda massivlarga o'xshash struktura **ro'yxat (list)** hisoblanadi. Pythonning o'zida aniq massiv tushunchasi yo'q, lekin ro'yxatlar massiv vazifasini bajaradi. Bundan tashqari, array moduli yoki NumPy kutubxonasi orqali aniq massivlarni ham ishlatish mumkin.

Massiv - bu bir xil turdagi ma'lumotlarni (masalan, sonlar, matnlar, obyektlar) ketma-ket saqlash uchun ishlatiladigan ma'lumotlar strukturasi. Massivlarning asosiy xususiyati shundaki, ularning barcha elementlari bir xil turga ega bo'ladi (masalan, faqat butun sonlar yoki faqat haqiqiy sonlar). Bu xususiyat massivlarni samarali va tezkor ishlash imkonini beradi.

Massivlarning har bir elementi indeks orqali murojaat qilinadi. Indekslar odatda 0 dan boshlanadi. Masalan, birinchi element indeksi 0, ikkinchi element indeksi 1 va hokazo.

Massivlarning asosiy xususiyatlari:

Bir xil turdagi ma'lumotlar: Massiv elementlari bir xil turdagi bo'lishi kerak (masalan, faqat butun sonlar yoki faqat matnlar).

Ketma-ket joylashgan xotira: Massiv elementlari xotirada ketma-ket joylashgan bo'ladi. Bu xususiyat massivlarga tez kirish imkonini beradi. Indeks orqali murojaat: Har bir elementga indeks orqali murojaat qilish mumkin.

O'lchami belgilangan: Massivning o'lchami (elementlar soni) odatda oldindan belgilanadi va o'zgartirilmaydi (lekin ba'zi tillarda dinamik massivlar mavjud).

Massivlarning afzalliklari:

Tez kirish: Massiv elementlariga indeks orqali tezda kirish mumkin. Bu operatsiya $O(1)$ vaqtda amalga oshiriladi.

Samaradorlik: Massivlar xotirada ketma-ket joylashganligi sababli, ular bilan ishlash juda samarali.

Oddiylik: Massivlarni tushunish va ishlatish juda oson.

Massivlarning kamchiliklari:

O'lchami cheklangan: Massivning o'lchami oldindan belgilanadi va uni o'zgartirish qiyin (lekin dinamik massivlar bu muammoni hal qiladi).

Elementlarni qo'shish/o'chirish qiyin: Massivga yangi element qo'shish yoki elementni o'chirish qiyin, chunki bu boshqa elementlarni siljitishni talab qiladi.

Xotira sarfi: Massivning o'lchami katta bo'lsa, u ko'p xotira egallashi mumkin.

Massivlarning turlari:

Bir o'lchamli massiv: Oddiy ro'yxat shaklida. Misol uchun: [1, 2, 3, 4, 5].

Ko'p o'lchamli massiv: Matritsa yoki kublar shaklida. Misol uchun, 2D massiv (matritsa):

```
[ [1, 2, 3],  
  [4, 5, 6],  
  [7, 8, 9]  
]
```

Pythonda massivlar (ro'yxatlar) bilan ishlash

Pythonda massivlarga eng yaqin struktura **ro'yxat (list)** hisoblanadi. Ro'yxatlar massivlarning barcha xususiyatlariga ega, lekin ular dinamik va turli turdagi ma'lumotlarni saqlashi mumkin.

Ro'yxat yaratish

```
mevalar = ["olma", "banan", "anor", "uzum"]
```

Elementga murojaat qilish

```
print(mevalar[0]) # "olma"
```

Ro'yxat uzunligi

```
print(len(mevalar)) # 4
```

```
# Ro'yxatga yangi element qo'shish
mevalar.append("nok")
print(mevalar) # ["olma", "banan", "anor", "uzum", "nok"]
```

```
# Ro'yxatdan elementni o'chirish
mevalar.remove("banan")
print(mevalar) # ["olma", "anor", "uzum", "nok"]
```

NumPy massivlari

Agar sizga aniq massivlar kerak bo'lsa (masalan, sonlar massivi), NumPy kutubxonasidan foydalanishingiz mumkin. Bu kutubxona yuqori samaradorlik bilan ishlaydi va katta ma'lumotlar to'plamlari bilan ishlashda qulay.

```
import numpy as np

# NumPy massivi yaratish
massiv = np.array([1, 2, 3, 4, 5])

# Massiv ustida amallar
print(massiv * 2) # [2, 4, 6, 8, 10]
print(massiv + 10) # [11, 12, 13, 14, 15]
```

Massivlarning xotirada joylashishi

Massiv elementlari xotirada ketma-ket joylashgan bo'ladi. Bu xususiyat massivlarga tez kirish imkonini beradi. Misol uchun, agar massivning birinchi elementi 0x1000 manzilda joylashgan bo'lsa, ikkinchi elementi 0x1004 manzilda joylashadi (4 bayt butun son uchun).

Massivlar — bu bir xil turdagi ma'lumotlarni ketma-ket saqlash uchun ishlatiladigan asosiy ma'lumotlar strukturasi. Ular tez kirish, samaradorlik va oddiylilik kabi afzalliklarga ega. Pythonda ro'yxatlar massiv vazifasini bajaradi, lekin aniq massivlar uchun NumPy kabi kutubxonalardan foydalanish mumkin.

Pythonda ro'yxatlar (massivlar) bilan ishlash

1. Ro'yxat yaratish

Ro'yxat kvadrat qavslar [] ichida elementlarni vergul bilan ajratib yoziladi.

```
python
mevalar = ["olma", "banan", "anor", "uzum"]
sonlar = [1, 2, 3, 4, 5]
```



```
aralash = [10, "salom", True, 3.14]
```

2. Ro'yxat elementlariga murojaat qilish. Ro'yxat elementlariga indeks orqali murojaat qilish mumkin. Indekslar 0 dan boshlanadi.

```
python  
print(mevalar[0]) # "olma"  
print(sonlar[2]) # 3
```

Agar indeks ro'yxat chegarasidan tashqarida bo'lsa, IndexError xatoligi yuzaga keladi.

3. Ro'yxatga yangi element qo'shish

append() metodi orqali ro'yxat oxiriga element qo'shish:

```
python  
mevalar.append("nok")  
print(mevalar) # ["olma", "banan", "anor", "uzum", "nok"]
```

insert() metodi orqali istalgan joyga element qo'shish:

```
python  
mevalar.insert(1, "gilos")  
print(mevalar) # ["olma", "gilos", "banan", "anor", "uzum", "nok"]
```

4. Ro'yxatdan element o'chirish

remove() metodi orqali qiymat bo'yicha elementni o'chirish:

```
python  
mevalar.remove("banan")  
print(mevalar) # ["olma", "gilos", "anor", "uzum", "nok"]
```

pop() metodi orqali indeks bo'yicha elementni o'chirish:

```
python  
mevalar.pop(2) # 2-indeksdagi "anor" ni o'chiradi  
print(mevalar) # ["olma", "gilos", "uzum", "nok"]
```

5. Ro'yxatni kesib olish (slicing)

Ro'yxatning bir qismini kesib olish uchun [:] sintaksisidan foydalaniladi.

```
python  
print(sonlar[1:3]) # [2, 3] (1-indeksdan 3-indeksgacha, 3 kirmaydi)  
print(sonlar[:3]) # [1, 2, 3] (boshidan 3-indeksgacha)  
print(sonlar[2:]) # [3, 4, 5] (2-indeksdan oxirigacha)
```

6. Ro'yxat uzunligi

Ro'yxat uzunligini len() funksiyasi orqali aniqlash mumkin.

```
python  
print(len(mevalar)) # 4
```

7. Ro'yxatni aylantirish

Ro'yxatni teskari tartibda chiqarish uchun[::-1] dan foydalaniladi.

```
python
print(sonlar[::-1]) # [5, 4, 3, 2, 1]
```

8. Ro'yxatni saralash
sort() metodi orqali ro'yxatni saralash:

```
python
sonlar.sort()
print(sonlar) # [1, 2, 3, 4, 5]
```

Teskari tartibda saralash:

```
python
sonlar.sort(reverse=True)
print(sonlar) # [5, 4, 3, 2, 1]
```

9. Ro'yxatni qo'shish
Ikki ro'yxatni + operatori yordamida birlashtirish mumkin.

```
python
yangi_mevalar = mevalar + ["anjir", "qovun"]
print(yangi_mevalar) # ["olma", "gilos", "uzum", "nok", "anjir", "qovun"]
```

10. Ro'yxatni takrorlash
Ro'yxat elementlari ustida for tsikli orqali yurish mumkin.

```
python
for meva in mevalar:
    print(meva)
```

NumPy massivlari

Agar sizga aniq massivlar kerak bo'lsa, NumPy kutubxonasidan foydalanishingiz mumkin. Bu kutubxona yuqori samaradorlik bilan ishlaydi va katta ma'lumotlar to'plamlari bilan ishlashda qulay.

```
python
import numpy as np

massiv = np.array([1, 2, 3, 4, 5])
print(massiv) # [1 2 3 4 5]
```

Grafiklar chizish

Python vektor va matritsalar ko'rinishidagi ma'lumotlarni, hisob-kitob natijalarini grafik ko'rinishda ifodalash va tasvirlash, shuningdek sharhlarni yaratish va shu grafiklarni saqlash uchun keng imkoniyatlarga ega. Ushbu paragrafda astronomiyada qo'llaniladigan ba'zi muhim grafik funksiyalar va ularning imkoniyatlari tasvirlangan, shuningdek, ularni

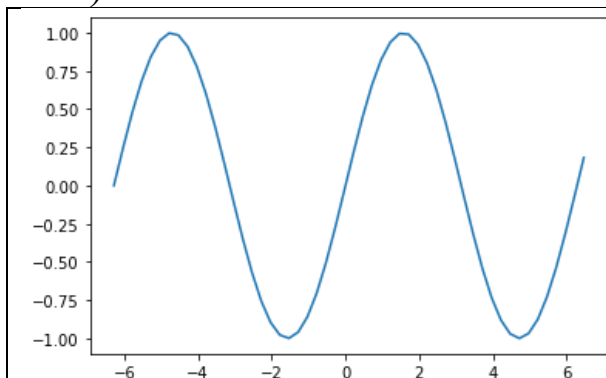
qo'llash misollari keltirilgan. Shu bilan birga, bu paragrafda biz ma'lumotlarni oddiy tekst fayllaridan o'qib olishni va qayta yozishni o'rganib olamiz.

Shunda eng ko'p qo'llaniladigan plot funksiyasidir. Lekin bunday funksiyadan foydalanish uchun matplotlib.pyplot kutubxonasiga murojlat qilish kerak bo'ladi. Uning argumentlari turli bo'lishi mumkin. x , y , t – biron bir kattaliklarning qiymatlari uchun vektor ko'rinishidagi massivlar bo'lsin. Agar grafik funksiyasini $\text{plot}(y)$ ko'rinishida bersak, unda y elementlarini indekslariga bog'langan chiziqli grafigini ko'ramiz. Agar bu funksiyani $\text{plot}(t,y)$ deb chaqirsak, unda y ning t ga nisbatan bog'liqligini ko'rsatadi.

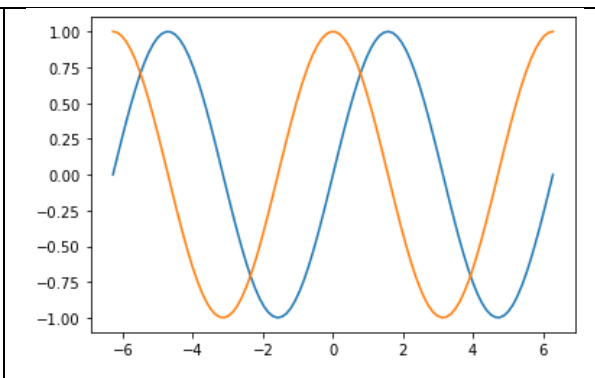
14-Misol. Oddiy trigonometrik funksiyalarning $[-2\pi; 2\pi]$ oraliqdagi grafiklarini ko'ramiz.

```
import numpy as np
t = np.linspace(-2* np.pi, 2* np.pi, 100)
# x va y massivni yaratib, bittasiga t ning Sinusini,
#% ikkinchisiga Cosinusini beramiz
x = np.sin(t); y = np.cos(t);
# cosinusning grafigini ko'ramiz
import matplotlib.pyplot as plt
plt.plot(t,x)
plt.plot(t,y)
```

Natijada **Grafiki** nomli oynada biz so'ragan grafikni ko'ramiz (6-rasm).



6-Rasm. $\text{plot}(t,y)$ bitta rasmda bitta funksiyani chizdi.



7-Rasm. Bitta grafikda bir nechta funksiya ko'rsatilgan.

Bitta grafikda bir nechta funksiya yoki bog'lanishni ko'rsatsa

bo‘ladi. Buni bir nechta yo‘l bilan amalga oshirsa bo‘ladi. Ikkalasining natijasi 7-rasmdagi grafik bo‘lib chiqadi.

$plot(t,x, t,y)$ $plot(t,x)$
 $plot(t,y)$

Ko‘rsatilganidek, turli bog‘lanishlarni ketma-ket vergul orqali kiritisa bo‘ladi (chap tomonda). Ammo bu yo‘l bilan ishlasak, quyida ko‘rsatilgan grafiklar parametrlarini boshqarish qiyin bo‘lib qoladi – yozuvlar juda uzun bo‘ladi va xatoliklar kirishi ehtimoli oshadi. Buning o‘rniga o‘ng tomonda ko‘rsatilgan hold on operatoridan foydalanamiz. U ketma-ket chiziqlarni bitta rasmda ko‘rsatishga imkon beradi. Unga teskari hold off operatori bu rejimni o‘chiradi (agarda buni yozmasak, chiziqlar ustma ust tushaveradi).

Bundan tashqari grid on operatori ham qo‘shildi. U, kerak bo‘lganda, grafikda xira rangdagi to‘rni hosil qiladi. Uni olib tashlash kerak bo‘lsa grid off deb yoziladi. Bu to‘r grafiklarda bormi yoki yo‘qligini visual ravishda ko‘rib olishga imkon beradi.

Yuqorida ko‘rsatilgan misollarda grafiklarning ranglari Spyder tomonidan avtomatik ravishda tanlanadi, chiziqlarning ko‘rinishi oddiy tekis o‘zgaradigan egri chiziq bilan ko‘rsatiladi. Ammo bu yerda qaysi grafik aynan qaysi funksiyaga tegishliligini bir nazardan aniqlab olish oson emas. Bu masalani osonlashtirish uchun grafikning parametrlarini (chiziq va nuqtalarning rangi va shakli, o‘qlari va yozuvlarning shrift va hokazolar) boshqarish kerak. Shunda grafiklarni o‘qish, tushunib olish ancha qulay bo‘ladi. Nimalarni boshqarsa va o‘zgartirsa bo‘ladi? Avval chiziqlarning shaklini ko‘ramiz

Chiziqning ko‘rinishi (Line Style)	Tavsifi	Ko‘rinishi
"_"	Uzluksiz chiziq	_____
"--"	Shtrixli chiziq	- - - - -
":."	Punktirli chiziq	
"-.""	Shtrix-punktirli chiziq	- - - - -

Keyin nuqtalarning shaklini belgilasa bo‘ladi.

Nuqtaning belgisi (Marker)	Tavsifi	Nuqtaning ko‘rinishi	Nuqtaning belgisi (Marker)	Tavsifi	Nuqtaning ko‘rinishi
"o"	Aylana	○	"+"	Plyus belgisi	+
"*"	Yulduzcha	*	"."	Nuqta	•
"x"	Xoch	×	"_"	Gorizontal chiziq	—
" "	Vertikal chiziq		"square"	Kvadrat	□
"diamond"	Romb	◇	"^"	Tepaga qaragan uchburchak	△
"v"	Pastga qaragan uchburchak	▽	">"	O‘ngga qaragan uchburchak	▷
"<"	Chapga qaragan uchburchak	◁	"pentagram"	Besh qirrali yulduz	☆
"hexagram"	Olti qirrali yulduz	☆			

Endi chiziq va nuqtalarning ranglarini ham belgilasa bo‘ladi.

Rang	Qisqacha belgisi	RGB uchtaligi	Nomi	Ko‘rinishi
"red"	"r"	[1 0 0]	Qizil	
"green"	"g"	[0 1 0]	Yasxil	
"blue"	"b"	[0 0 1]	Ko‘k	
"cyan"	"c"	[0 1 1]	Moviy	
"magenta"	"m"	[1 0 1]	Binafsha	
"yellow"	"y"	[1 1 0]	Sariq	
"black"	"k"	[0 0 0]	Qora	
"white"	"w"	[1 1 1]	Oq	

Rang	Qisqacha belgisi	RGB uchtaligi	Nomi	Ko‘rinishi
"none"	Rang umuman yo‘q bo‘lgan hollarda belgilanadi. Belgi, nomi, ko‘rinishi va boshqalari mavjud emas			

Keyingi masala kelib chiqadi – bu parametrlar qanday kiritiladi? Buning ham bir necha yo‘li bor. Bittasida hamma parametrlarni plot funksiyasi ichida qo‘shimcha argument sifatida bersa bo‘ladi:

```
plot(t,x,'b--d')
```

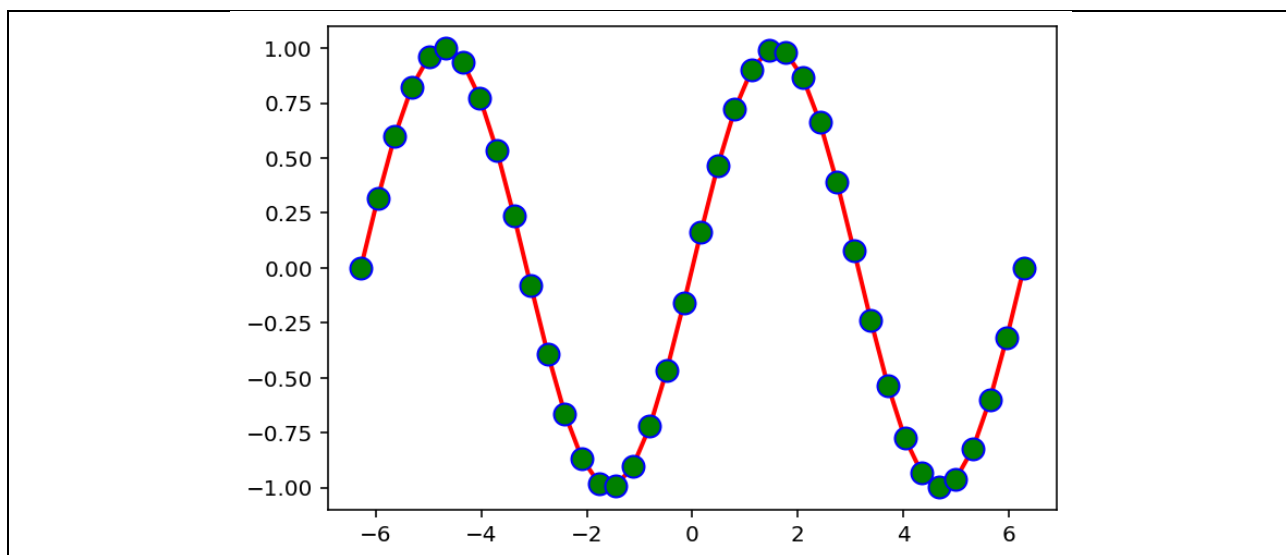
```
hold on
```

```
plot(t,y,'r-o')
```

Bu yerda ‘b--d’ va ‘r-o’ chiziqlarning rangini, ko‘rinishi hamda nuqtaning shakli ko‘rsatilgan. Aynan shu tartibda bo‘lishi shart emas. Lekin grafiklarning argumentlarini shu tartibda beradigan bo‘lsa, buyruqlarimiz juda uzun bo‘lib ketishi mumkin, chunki chiziq, nuqtalar, yozuvlarning parametrlarini ham o‘zgartirish kerak bo‘lib qolishi mumkin. Parametrlarni alohida belgilash yo‘li bor:

15-Misol. Grafikni chizish

```
import numpy as np
t = np.linspace(-2* np.pi, 2* np.pi, 40)
# x va y massivni yaratib, bittasiga t ning Sinusini,
#% ikkinchisiga Cosinusini beramiz
x = np.sin(t); y = np.cos(t);
import matplotlib.pyplot as plt
line, = plt.plot(t,x) ;
line.set_color('r'); #Chiziqning rangi belgilandi
line.set_linestyle('-'); #Chiziqning ko‘rinishi belgilandi
line.set_linewidth(2); #Chiziqning qalinligi belgilandi
line.set_marker('o'); #Nuqtaning shakli belgilandi
line.set_markersize(10); #Nuqtaning o‘lchami belgilandi
line.set_machedgecolor('b'); #Nuqtaning chetki rangi belgilandi
line.set_markerfacecolor('g'); #Nuqtaning ichki rangi belgilandi
plt.show()
```



Bu bilan biz grafikni ixtiyoriy talab etilgan shaklga keltirishimiz mumkin. Rangga kelsak, uning xilma-xilligi ko'rsatilgan 7 rang bilan chegaralanmaydi. RGB uchtaligi yordamida xohlagan rangni belgilasa bo'ladi. RGB uchtaligi – bu uch elementli qatorli vektordir. Uning elementlari rangning qizil, yashil va ko'k qismlarining intensivliklarini belgilaydi. Intensivlik $[0,1]$ oralig'ida bo'lishi kerak. Shunda rangni, masalan, quyidagicha ham belgilasa bo'ladi: `g.Color = [1 0 0.2]`; `g.MarkerEdgeColor = [0.2 0.1 1]`; `g.MarkerFaceColor = [0.73 1 0.36]`; Endi grafikning o'qlarini ko'rsatamiz, kerak bo'ladigan yozuvlarni kiritamiz.

16-Misol. Grafik funksiyaning parametrlarini belgilash

```
import numpy as np
t = np.linspace(-2* np.pi, 2* np.pi, 60)
# x va y massivni yaratib, bittasiga t ning Sinusini,
# ikkinchisiga Cosinusini beramiz
x = np.sin(t); y = np.cos(t);
import matplotlib.pyplot as plt
g, = plt.plot(t,x,label='Sin(t)') ; p, = plt.plot(t,y,label='Sos(t)') ;
#Chiziqning rangi belgilandi
g.set_color((1, 0, 0.2)); p.set_color((0, 0.7, 1));
g.set_linestyle('-'); p.set_linestyle('-') #Chiziqning ko'rinishi belgilandi
g.set_linewidth(2); p.set_linewidth(2) #Chiziqning qalinligi belgilandi
g.set_marker('o'); p.set_marker('s') #Nuqtaning shakli belgilandi
g.set_markersize(7); p.set_markersize(7) #Nuqtaning o'lchami
belgilandi
#Nuqtaning chetki rangi belgilandi
g.set_machedgecolor((0.2, 0.1, 1)); p.set_machedgecolor('r')
#Nuqtaning ichki rangi belgilandi
```



```
g.set_markerfacecolor((0.73, 1, 0.36)); p.set_markerfacecolor('y')
```

```
#Chiziqlar yonida izohlar kiritiladi
```

```
plt.text(2.85, 0.4, ' sin(t)', fontsize=12)
```

```
plt.text(-2.5, 0.4, ' cos(t)', fontsize=12)
```

```
#Chiziqlarning alohoda izohi kiritiladi
```

```
plt.legend()
```

```
#Grafikda to'r kiritildi
```

```
plt.grid(True, alpha=0.5)
```

```
#Grafikni tushuntiradigan yozuvlar kiritiladi
```

```
plt.title('Trigonometrik funksiyalar');
```

```
plt.xlabel('Vaqt (t)');plt.ylabel('SIN(t), COS(t)');
```

```
#O'qlarning o'lchamlari belgilanadi
```

```
plt.axis((-5, 5, -1, 1));
```

```
#O'qlardagi yozuvlarning o'lchami boshqariladi
```

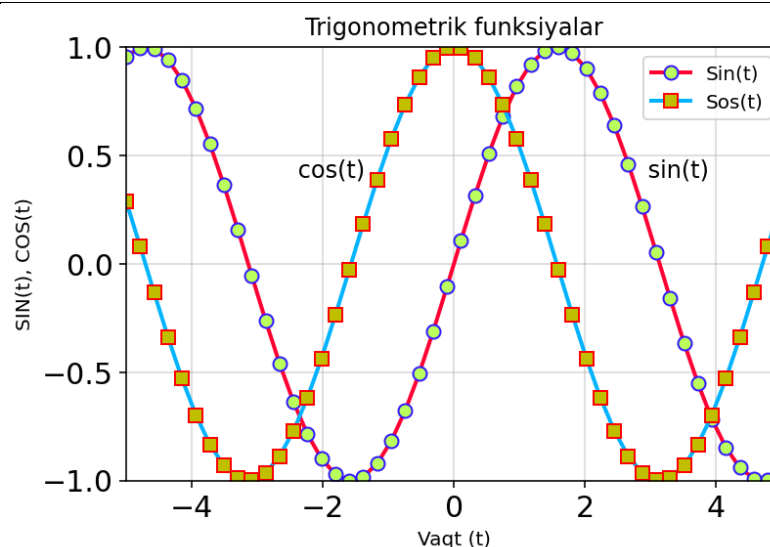
```
ax = plt.gca(); ax.tick_params(axis='both', labelsz=15)
```

```
# Grafikni ko'rsatilgan formatda saqlasa bo'ladi
```

```
plt.savefig('plot.png', bbox_inches='tight', dpi=600)
```

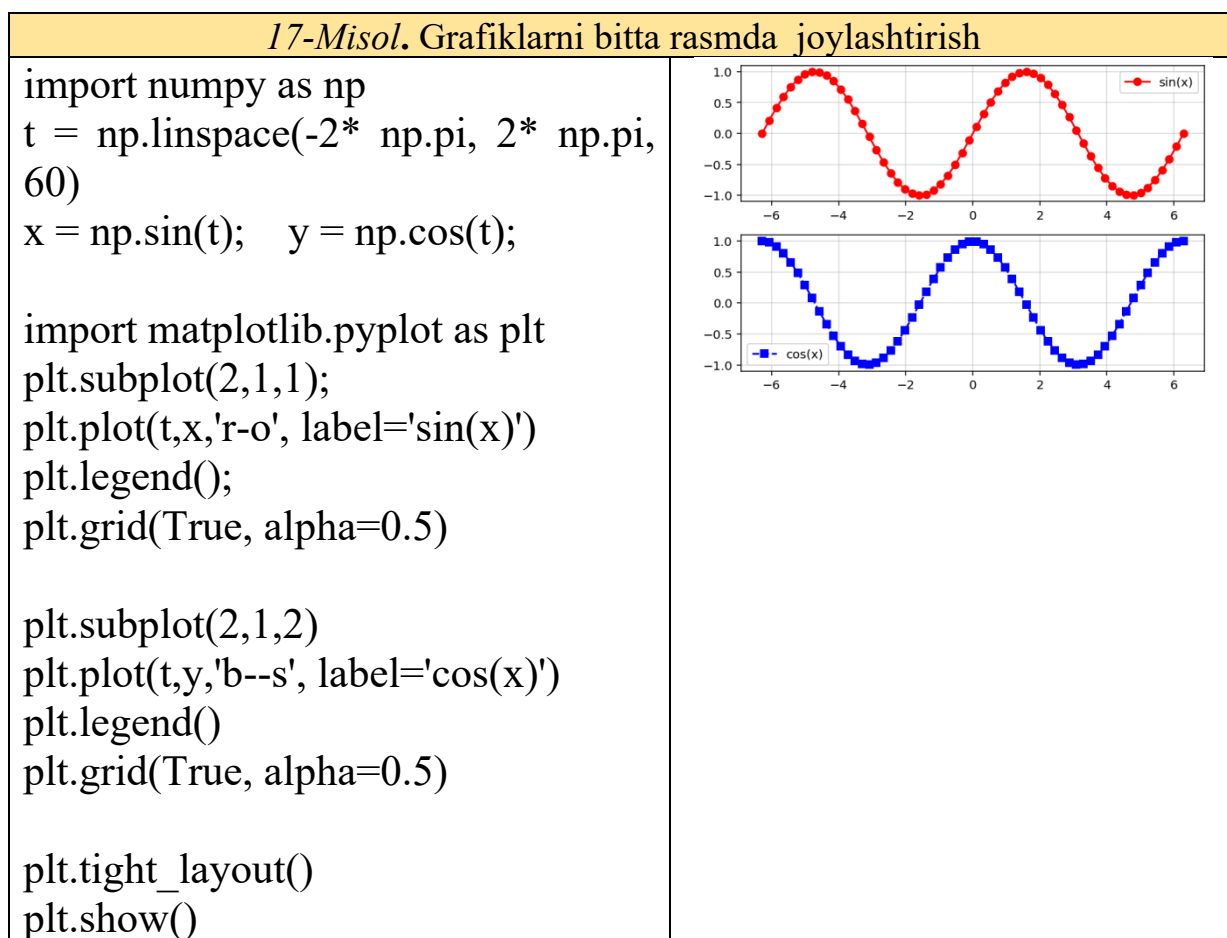
```
plt.savefig('plot.jpg', bbox_inches='tight', dpi=600) # Grafikni JPEG  
formatida saqlab beradi
```

```
plt.savefig('plot.pdf', bbox_inches='tight', dpi=600) # Grafikni PDF  
formatida saqlab beradi
```



xlabel('Vaqt (t)') va ylabel('SIN(t), COS(t)') funksiyalar yordamida gorizontaal va vertikal o'qlarda yotgan kattaliklar haqida ma'lumot beramiz. title('Trigonometrik funksiyalar') deb grafik haqida umumiy

ma'lumot beriladi. `legend('sin(t)', 'cos(t)')` – bu funksiyadagi ma'lumotlar har bir chiziq nima ekanligini oydinlashtiradi. `axis([-5 5 -1 1])` – gorizontaal va vertikal o'qlarning quyi va yuqori chegaralarini belgilaydi. Misolni tahlil qilganda boshqacha qiymatlar berib, nima bo'lishini ko'rsangiz bo'ladi. `text(pi,0, 'sin(\pi)', 'FontSize', 18)` funksiyasi yordamida grafiklarga tekst ko'rinishidagi izohlarni kiritisa bo'ladi. Bu yerda `pi,0` – mos ravishda gorizontaal va vertikal o'qlar bo'yicha koordinatalar, `'sin(\pi)'` – o'rnatilishi kerak bo'lgan yozuv, `'FontSize', 18` – shrift kattaligi. `ax = gca; ax.FontSize = 16` parametrlar yordamida o'qlardagi shriftlar boshqariladi.



Endi turli bog'liqliklarni bitta grafikka emas, balki bitta rasmda joylashgan bo'lishi kerak degan talab etilsa, `subplot(n,m,p)` komandasidan foydalanamiz (17-Misolga qara). Bu yerda `n` – satrlar soni, `m` – ustunlar soni, `p` – tartib raqami.

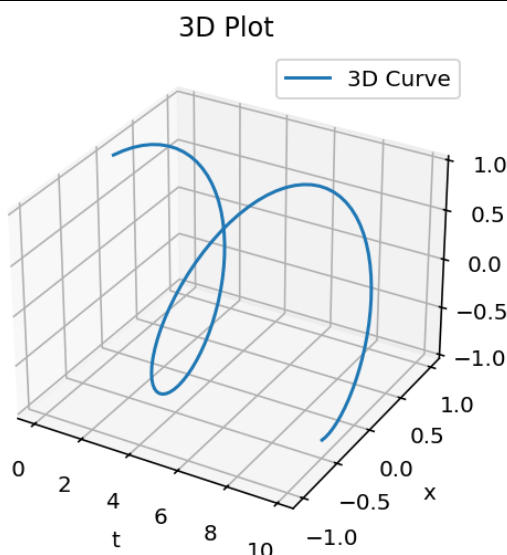
Keyinchlik, differensial va harakat tenglamalarni sonli yechganimizda, natijalarni tahlil qilish uchun sinov nuqta yoki jismning harakatini ko'rishimiz kerak bo'ladi. Bunda `comet` yoki `addpoints` komandalaridan foydalanamiz.

Ikki o'lchamli grafiklar kabi uch o'lchamli grafiklarni ham chizamiz.

19-Misol. Ikki o'lchamli grafiklar

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D # Импорт для 3D-
графиков

# Создаем данные
t = np.arange(0, 10, 0.1)
x = np.sin(t); y = np.cos(t)
# Создаем 3D-график
fig = plt.figure() # Создаем фигуру
ax = fig.add_subplot(111, projection='3d') # Добавляем 3D-
подграфик
# Строим график
ax.plot3D(t, x, y, label='3D Curve') # Аналог plot3 в MATLAB
# Добавляем подписи осей и заголовков
ax.set_xlabel('t'); ax.set_ylabel('x')
ax.set_zlabel('y'); ax.set_title('3D Plot')
ax.legend()
plt.show()
```



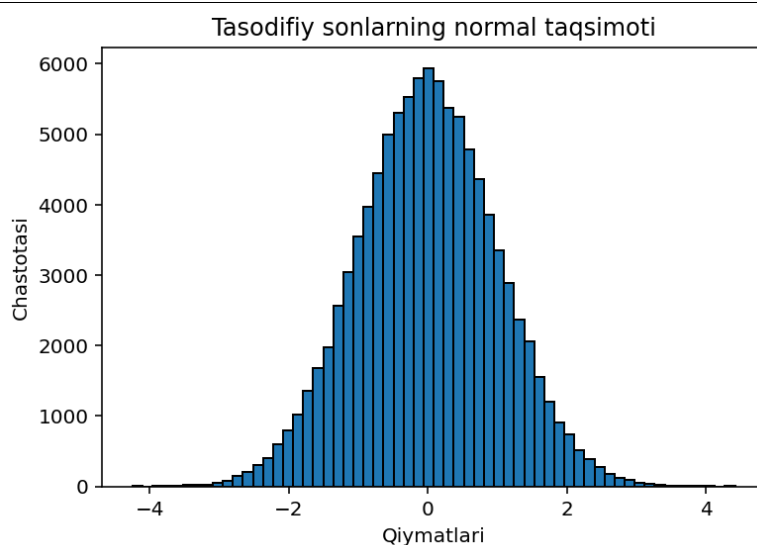
Astronomiyada gistogrammalar ham ko'p ishlatiladi. Gistogramma bu – biron bir kattalikning qiymatlari bo'yicha taqsimlanganligi ko'rsatadigan grafik. Boshqacha qilib aytganda, biron bir kattalikning ehtimolligi zichligidir. Masalan, normal taqsimlangan tasodifiy sonlarning

gistogrammasini yarating, degan masala qo'yildi. Tasodifiy sonlarni generator komandasi yordamida hosil qilamiz, natijasini gistogramma ko'rinishiga olib kelamiz.

20-Misol. Gistogrammalar uchun misollar

```
import numpy as np
import matplotlib.pyplot as plt

# Tasodifiy sonlarni generatsiyasi
y = np.random.randn(100000)
# Gistogrammani chizish
plt.hist(y, bins=60, edgecolor='black')
# Izohlarni kiritamiz
plt.xlabel('Qiymatlari'); plt.ylabel('Chastotasi')
plt.title('Tasodifiy sonlarning normal taqsimoti')
# Grafikni ko'rsatamiz
plt.show()
```



NAZORAT SAVOLLARI

1. Massivning izohini keltiring. Massivning elementlari qanday chaqiriladi?
2. Pythonida birinchi elementning indeksi nechiga teng?
3. Massivning asosiy xususiyatlarini sanab bering
4. Bitta massivda turli tipdagi elementlar bo'lishi mumkinmi?
5. Qaysi funksiya yordamida massivga yangi elementi qo'shsa bo'ladi?
6. Qaysi funksiya yordamida massivga yangi elementi olib tashlasa bo'ladi?
7. len() funksiyasi qanaqa kattalikni beradi?

8. Qaysi funksiya yordamida massivning ixtiyoriy joyiga yangi elementno qo'shsa bo'ladi ?
9. Qaysi funksiya yordamida massivning oxiriga yangi elementno qo'shsa bo'ladi ?
10. `A.insert(n,b)` komandasi nimani amalga oshiradi?
11. `import matplotlib.pyplot` buyrug'i nimani bajaradi?
12. `plt.plot(x, np.sin(x), 'r-o')` komandasi grafik chizadi. Uning rangi, ko'rinishi va nuqtakarning shakli qanaqa bo'ladi?

4-MAVZU: INTEGRALLASH HAMDA ODDIY DIFFERENSIAL TENGLAMANI YECHISH SONLI USULLARI

REJA

5. Sonli integrallash usullari algiritlemlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algiritlemlari.
6. Ular yordamida aniq integrallarni hisoblash.
7. Oddiy differensial tenglamalarni sonli yechish usullari va algiritlemlari.
8. Mexanik harakatlarni modelashtirish

Integrallarni hisoblash usullari

Integrallash matematikada va fizikada juda muhim hisoblash usuli hisoblanadi. Uning asosiy maqsadi, biror miqdorning o'zgarishini hisoblash yoki biror maydon, hajm, ish, energiya kabi fizik miqdorlarni hisoblashda qo'llaniladi.

Integrallash (integral) matematikada funksiyaning ma'lum bir oraliq bo'yicha yig'indisini hisoblash jarayoni hisoblanadi. U differensiallashning teskarisi sifatida ham talqin qilinadi. Integrallashning asosiy maqsadi, funksiya ostidagi maydonni, hajmni, yoki biror uzluksiz miqdorni hisoblashdir.

Integrallashning rasmiy ta'rif: *Integrallash, funksiyaning ma'lum bir oraliqda (masalan, a dan b gacha) qiymatlarini yig'ish jarayoni hisoblanadi.* Bu jarayonni quyidagicha ifodalash mumkin:

$$\int_a^b f(x)dx,$$

bu yerda: $f(x)$ – integrallanadigan funksiya,
 a va b — integrallash chegaralari (pastki va yuqori chegaralar),
 dx – o'zgaruvchining differensial (masalan, x o'zgaruvchisining kichik o'zgarishi).

Integrallashning fizikadagi mohiyati: *Integrallash, asosan, biror miqdorning kichik qismlarini yig'ish orqali umumiy qiymatni topish jarayoni hisoblanadi.*

Fizikada bu jarayon ko'pincha quyidagi holatlarda qo'llaniladi:

1. Maydon yoki hajmni hisoblash:

Masalan, biror egri chiziq ostidagi maydonni hisoblashda integrallash qo'llaniladi. Agar sizda $f(x)$ funksiya bo'lsa, uning a dan b gacha bo'lgan integrali shu funksiya ostidagi maydonni beradi:

$$\text{Maydon} = \int_a^b f(x) dx,$$

2. Ish va energiya:

Fizikada ish, kuch va masofa orqali hisoblanadi. Agar kuch $F(x)$ masofaga bog'liq bo'lsa, u holda a dan b gacha bo'lgan masofada bajarilgan ish:

$$Ish = \int_a^b F(x) dx,$$

Xuddi shu tarzda, energiyani ham integrallash orqali hisoblash mumkin.

3. Tezlik va masofa:

Agar sizda tezlik $v(t)$ vaqtga bog'liq bo'lsa, u holda t_1 dan t_2 gacha bo'lgan vaqt oralig'ida bosib o'tilgan masofa:

$$Masofa = \int_{t_1}^{t_2} v(t) dt,$$

4. Zaryad va tok:

Elektr zaryadi q va tok $I(t)$ orasidagi bog'lanish:

$$q = \int I(t) dt$$

bu yerda tok vaqt bo'yicha integrallanib, zaryadni beradi.

Misol: Aytaylik, sizda bir jismning tezligi $v(t) = 3t^2 + 2t$ (bu yerda t vaqt) bo'lsin. Siz 0 dan 5 soniyagacha bo'lgan vaqt oralig'ida bosib o'tilgan masofani topmoqchisiz. Buning uchun tezlikni vaqt bo'yicha integrallaysiz:

$$Masofa = \int_0^5 (3t^2 + 2t) dt,$$

bu integralni hisoblab, masofani topishingiz mumkin.

Integrallash fizikada biror miqdorning kichik qismlarini yig'ish orqali umumiy qiymatni topish uchun ishlatiladi. Bu jarayon maydon, ish, energiya, masofa va boshqa ko'plab fizik miqdorlarni hisoblashda qo'llaniladi.

Integral bu matematik tahlilning eng muhim tushunchalaridan biri. Integral egri chiziqli ostidagi maydonni topish muammolarini hal qilishda, notekis harakat bilan bosib o'tgan yo'lni hisoblashda, funktsiyani uning hosilasi bo'yicha tiklash masalalarida, bir jinsli bo'lmagan jismlarning massalari aniqlash va boshqa masalalarni yechishda keng qo'llaniladi. Soddalashtirilgan holda, integralni cheksiz sonli cheksiz kichik hadlarning yig'indisi sifatida ifodalash mumkin. Sonli usullar sohasida odatda aniq integrallar bilan ish tutiladi:

$$F(x) = \int_a^b f(x) dx,$$

bu yerda $f(x)$ – integral ostidagi funksiya, $[a, b]$ – integrallash chegaralari.

Spyderda sistemasida aniq va noaniq integrallarni yechish uchun turli

imkoniyatlar mavjud. Integrallarni sonli usullar bilan hisoblaganda integral ostidagi funksiyani iloji boricha soda ko‘rinishga keltirish kerak bo‘ladi, chunki bu hisob-kitoblarni soddalashtiradi. Qaysi usul bilan ishlashni odatda foydalanuvchining o‘zi tanlaydi. Bu usullarning hammasi taqribiy qiymatlarni berishini esdan chiqarmaslik kerak.

Integrallarni hisoblash uchun qator sonli formulalar mavjud – to‘g‘riburchaklar, trapetsiyalar, parabolalar (Simpson) usullari. Ularni bo‘yicha hisoblash juda oson – oddiy sikl yordamida amalga oshirsa bo‘ladi.

Xuddi differensial tenglamalardagi misollardek, quyidagi misollarni shablon sifatida ishlatas bo‘ladi – qizil rang bilan belgilangan yozuvlarni masalaga moslashtirilsa bo‘ldi.

To‘g‘ri burchaklar formulasida integrallash $[a, b]$ diapazoni n ta bo‘lakka bo‘linadi va funksiyaning k -inchi ($k = 1, 2, \dots, n$) tugunlardagi y_k qiymatlarinig yig‘indisi olinadi. Oxirida summa $h = (b - a)/n$ integrallash qadamiga ko‘paytiriladi:

$$\int_a^b f(x) dx = h \sum_{k=1}^n y_k$$

$\int_1^2 \frac{dt}{t^3} dt$	$\int_{0.6}^{\pi/3} \sin x dx$
<pre>import numpy as np import math as mt # Integrallash diapazoni a = 1; b = 2; # Bo‘laklar soni va integrallash qadami n = 1000; h=(b-a)/n; # Argumentnig tug‘unlardagi qiymatlari t = np.linspace(a, b, n) # Summaning birinchi hadi S = 0; for k in range (1,n): # Integral ostidagi funksiya S = S + 1 / mt.pow(t[k],3); S = S * h; print('\n Integral S = %2.4f ga teng \n' % S)</pre>	<pre>import numpy as np import math as mt # Integrallash diapazoni a = 0.6; b = mt.pi/3; # Bo‘laklar soni va integrallash qadami n=1000; h=(b-a)/n; # Argumentnig tugunlardagi qiymatlari y = np.linspace(a, b, n) # Summaning birinchi hadi S = 0; for k in range(1,n): # Integral ostidagi funksiya S = S + mt.sin(y[k]); S = S * h; print('\n Integral S = %2.4f ga teng \n' % S) # Integral S = 0.3253 ga teng</pre>

Integral S = 0.3750 ga teng

To'g'ri burchaklar formulasida integrallash $[a, b]$ diapazoni n ta bo'lakga bo'linadi va funksiyaning tugunlardagi y_k qiymatlarinig yig'indisi olinadi, faqat birinchi va oxirgi hadlarning yarmi olinadi. Oxirida summa integrallash qadami h ga ko'paytiriladi:

$$\int_a^b f(x) dx = h \left(\frac{y_1 + y_n}{2} + \sum_{k=2}^{n-1} y_k \right)$$

$y_1 = f(a)$ va $y_n = f(b)$ – integral osti funksiyaning integrallash diapazonining chegaralaridagi qiymatlari.

$\int_1^5 \frac{1}{x} dx$	$\int_0^1 \frac{1}{1+x^2} dx$
<pre>import numpy as np import math as mt a=1; b=5; n=1000000; h=(b-a)/n; y = np.linspace(a, b, n) S = 0; for k in range(2,n-1): S = S + 1 / y[k]; S = S + (y[0] + y[n-1]) / 2; S = S * h; print('\n Integral S = %2.4f ga teng \n' % S) # Integral S = 1.6096 ga teng</pre>	<pre>import numpy as np import math as mt a = 0; b = 1; n = 10000; h = (b-a) / n; y = np.linspace(a, b, n) S = a; for k in range (1,n-1): S = S + 1 / (1 + mt.pow(y[k],2)) S = S + (y[0] + y[n-1]) / 2; S = S * h; print('\n Integral S = %2.4f ga teng \n' % S) # Integral S = 0.7853 ga teng</pre>

Simpson formulasida toq indeksli hadlari 4 ko'paytiriladi, juft indeksli hadlar 2 ko'paytiriladi, birinchi va oxirgisi o'zgarmaydi. Hammasi yig'iladi hamda integrallash qadamining uchda biriga ko'paytiriladi:

$$\int_a^b f(x) dx = \frac{h}{3} (y_1 + 2y_2 + 4y_3 + 2y_4 + 4y_5 + \dots y_n)$$

$\int_0^3 \sqrt[3]{x} dx$	$\int_0^{\pi/6} \sqrt{\tan x} dx$
<pre> import numpy as np import math as mt a = 0; b = 3; n = 1000000; h = (b-a) / n; y = np.linspace(a, b, n) S1 = 0; S2 = 0; for k in range(1,n-1,2): S1 = S1 + mt.pow(y[k],3/2) * 2; for k in range (2,n-1,2): S2 = S2 + mt.pow(y[k],3/2) * 4; S = (S1 + S2 + y[0] + y[n-1]) * h / 3; print('\n Integral S = %2.4f ga teng \n' % S) # Integral S = 6.2354 ga teng </pre>	<pre> import numpy as np import math as mt a = 0; b = mt.pi/6; n = 10000; h = (b-a) / n; y = np.linspace(a, b, n) S1 = 0; S2 = 0; for k in range(1,n-1,2): S1 = S1 + mt.sqrt(mt.tan(y[k-1])) * 2; for k in range(3,n-1,2): S2 = S2 + mt.sqrt(mt.tan(y[k])) * 4; S = (S1 + S2 + y[0] + y[n-1]) * h / 3; print('\n Integral S = %2.4f ga teng \n' % S) #Integral S = 0.2578 ga teng </pre>

Integrallarni hisoblash uchun MATLABda oʻrnatilgan operatorlar mavjud. Ular orasida nisbatan oson qoʻllaniladigani int funksiyasi. Uni qanday qoʻllanilishini koʻrib chiqamiz:

Integral koʻrinishi
$\int_0^{\pi/3} \sin 2x dx$
Yechimi
<pre> import numpy as np from scipy.integrate import simpson # Integrallash kerak boʻlgan funktsiyani belgilaymiz def f(x): return np.sin(2*x) </pre>

<pre> # Integrallash diapazonini hamda nuqtalar sonini belgilaymiz a = 0 # Quyi chegara b = np.pi/3 # Yuqori chegara n = 100 # Nuqtalar soni x = np.linspace(a, b, n) # Nuqtalar bir tekis taqsimlanad y = f(x) # Funksiyaning bu nuqtalardagi qiymati # Integral Simpson usuli bilan hisoblanadi S = simps(y, x) print('\n Integral S = %2.4f ga teng \n' % S) </pre>
Javobi
Integral S = 0.7500 ga teng
$\int_0^{\infty} e^{-x^2} dx$
<pre> import numpy as np from scipy.integrate import quad # Integrallash kerak bo'lgan funktsiyani ko'rsatamiz def f(x): return np.exp(-x**2) # Integralni 0 dan +∞ gacha hisoblaymiz S, error = quad(f, 0, np.inf) print('\n Integral S = %2.4f ga teng \n' % S) </pre>
Javobi
Integral S = 0.8862 ga teng
Spyder bilan nafaqat aniq integrallarni, balki simvolli amallarni ham bajarsa bo'ladi. Ya'ni integrallarni analitik ko'rinishdagi yechimlarini topish mumkin. Masalan:
$\int (x^2 + a) dx$
<pre> import sympy as sp # Simvilli o'zgaruvchanlarni belgilaymiz x = sp.symbols('x') a = sp.symbols('a') # Doimiy kattalik (parameter) # Funktsiyani ko'rsatamiz f = x**2 + a # Noaniq integral hisoblanadi integral = sp.integrate(f, x) print("Noaniq integral :", integral) </pre>
Javobi
Noaniq integral : a*x + x**3/3

Integral aniq qo'rinishda bo'lsa

$$\int (x^2 + a) dx$$

Unda yana bir nechta qator qo'shiladi

```
import sympy as sp
# Simvilli o'zgaruvchanlarni belgilaymiz
x = sp.symbols('x')
a = sp.symbols('a') # Doimiy kattalik (parameter)
# Funksiyani ko'rsatamiz
f = x**2 + a
# Noaniq integral hisoblanadi
integral = sp.integrate(f, x)
print("Noaniq integral :", integral)
```

```
a1 = 0
b1 = 2*sp.pi
```

```
# Aniq integral hisoblanadi
definite_integral = sp.integrate(f, (x, a1, b1))
print("Aniq integral:", definite_integral)
```

Javobi

```
# Noaniq integral : a*x + x**3/3
# Aniq integral: 2*a + 8/3
```

Differensial tenglamalarni yechish

Differensial tenglama – bu mustaqil o'zgaruvchanlar va ushbu o'zgaruvchanlarning noma'lum funksiyalaridan tashqari, shu noma'lum funksiyalarning hosilalarini ham o'z ichiga olgan tenglamaga aytamiz. Boshqacha qilib aytganda, funksiya $y(t)$, uning argumenti t hamda ushbu funksiyaning $y', y'', \dots, y(n)$ hosilalarini bir-biri bilan ulaydigan tenglama differensial tenglama deyiladi. Umumiy holda differensial tenglamani quyidagi ko'rinishda yozsa bo'ladi:

$$F(t, y', y'', \dots, y(n)) = 0$$

Berilgan ixtiyoriy t nuqtada funkfiysning $y(t)$ qiymatini topish kerak degan masala qo'yiladi

Agarda tenglamada ishtirok etayotgan noma'lum bo'lgan $y = y(t)$ funksiya bitta argunmetga bog'langan bo'lsa, unda differensial tenglama *oddiy differensial tenglama* deyiladi. Agar tenglama bir nechta mustaqil o'zgaruvchilarga nisbatan noma'lum funksiyalarning xususiy hosilalarini o'z ichiga olsa, u holda tenglama *xususiy differensial tenglama* deb ataladi.

Differensial tenglamaga kiritilgan noma'lum funksiya hosilalarining eng yuqori tartibi *differensial tenglamaning tartibi* deyiladi. Birinchi tartibdagi differensial tenglamaning ko'rinishi $F(t, y, y') = 0$. Shunda birinchi tartibdagi tenglama funksiyaning hosilasig nisbatan quyidagicha yoziladi:

$$y' = f(t, y) \Leftrightarrow \frac{dy}{dt} = f(t, y)$$

Ikkinchi tartibdagisi esa $F(t, y, y', y'') = 0$

$$y'' = f(t, y, y') \Leftrightarrow \frac{d^2y}{dt^2} = f(t, y, y').$$

Biz bu yerda faqat oddiy differensial tenglamalarni ko'ramiz. Shu bilan birga, astronomik va fizik jarayonlarning dinamikasini tavsiflashda ushbu jarayonlarni xarakterlovchi barcha kattalilar vaqtga bog'liq bo'lganligi sababli mustaqil o'zgaruvchini t bilan belgilaymiz.

Fizikadagi ko'plab masalalar differensial tenglamalarni yechishga olib keladi. Buning sababi shundaki, fizik jarayonlarni tavsiflovchi deyarli barcha fizik qonunlar turli funksiyalarga nisbatan differensial tenglamalardir. Ushbu fizik qonunlar ko'plab tajribalarning nazariy umumlashtirilishini ifodalaydi va o'rganilayotgan kattaliklarni umumiy holda ham fazodagi ham vaqt davomidagi o'zgarishini ko'rsatadi. Xususan, Nyutonning ikkinchi qonuni ikkinchi tartibdag differensial tenglama:

$$m \frac{d^2r}{dt^2} = F(t, r, v,) \quad (1)$$

Shunda dinamikaning asosiy masalasi dinamikaning asosiy masalasi moddiy nuqtaning F kuchi ta'sirida harakat qonunini $r = r(t)$ aniqlashdir. Bu esa matematik jihatda (25) tenglamaning yechimini bildiradi va shu nuqtaning koordinatalarini vaqtga nisbatan funksiyalarini $x = x(t), y = y(t), z = z(t)$ belgilaydi.

Odatda, yuqoridagi ko'rinishdagi tenglamalar cheksiz sonli yechimlarga ega bo'ladi $y = y(t)$. U ular orasida yagona, aniq, xususiy holdagi bittasini ajratib olish uchun yechimda qo'shimcha talablar qo'yiladi. Ko'p hollarda Koshi masalasi qo'yiladi: differensial tenglamaning $y = y(t)$ yechimlari orasida shunday yechimni topish kerakki, u berilgan argumentda aniq qiymatga ega bo'lishi kerak. Ya'ni $t = t_0$ bo'lganda

$$y(t_0) = y_0 \quad (2)$$

tenglik haqiqiy bo'ladi. Shunda (t_0, y_0) – boshlang'ich qiymatlari deyiladi, (2)-shart esa boshlang'ich sharti deyiladi.

Fizika va asrofizikani o'rganish uchun differensial tenglamalarning ahamiyatini hisobga olgan holda amalda uchraydigan differensial tenglamalarning ayrim turlarini va ularni yechish usullarini aniq misollarda ko'rib chiqamiz. Ushbu qo'llanmaning asosiy vazifasi astronomik masalalarni kompyuter yordamida yechishni o'rgatish bo'lgani uchun, biz e'tiborimizni differensial tenglamalarni sonli yechish usulariga qaratamiz.

Eyler usuli bilan birinchi tartibdagi differensial tenglamani yechib ko'ramiz. Avval differensialni analitik ko'rinishidan diskret ko'rinishiga olib kelimiz

$$\frac{dy}{dt} \approx \frac{\Delta y}{\Delta t} = f(t, y)$$

$\Delta y = y_1 - y_0$ va $\Delta t = t_1 - t_0 = h$ desak, (27) qayta yoziladi

$$\frac{y_1 - y_0}{h} = f(t_0, y_0)$$

Bu yerda h - hisob-kitob olib boriladigan vaqt o'qi bo'yicha qadam, odatda u quyidagicha topiladi: $h = (t - t_0)/n$ (t_0 - boshlang'ich vaqt, t - chegaraviy vaqt, n - bo'laklar soni).

(1)-dan 1-qadam uchun formula kelib chiqadi:

$$\begin{cases} y_1 = y_0 + h * f(t_0, y_0) \\ t_1 = t_0 + h \end{cases}$$

Uni umumlashtiradigan bo'lsak, quyidagiga kelimiz:

$$\begin{cases} y_n = y_{n-1} + h * f(t_{n-1}, y_{n-1}) \\ t_n = t_{n-1} + h \end{cases}$$

Ko'rib turibmizki, oddiy differensial tenglama chiziqli tenglamalar sistemasi ko'rinishigacha soddalashtirildi. Bu tenglamalar sistemasi yordamida Koshi masalasini bimalol yechsa bo'ladi. Agarda tenglama ikkinchi tartibli bo'lsa, (28) ga yana bitta had qo'shiladi:

$$\begin{cases} y'_n = y'_{n-1} + h * f(t_{n-1}, y_{n-1}, y'_{n-1}) \\ y_n = y_{n-1} + h * y'_n \\ t_n = t_{n-1} + h \end{cases}$$

Runge-Kutta usuli

Vaqt bo'yicha qadamni belgilaymiz. Uning uchun vaqt bo'yicha berilgan intervalni mayda qadamlarga bo'lamiz. Oldingi misoldek: $\Delta t = h = (t - t_0)/n$, bu yerda t_0 - boshlang'ich shartidagi vaqt, t - chegaraviy vaqt, n - bo'laklar soni.

Birinchi tartibdagi differensial tenglama uchun $y' = f(t, y)$ $t_k = t_0 + k \cdot h$ $k = 1, 2, \dots, n$
$y_{k+1} = y_k + (p_1 + 2p_2 + 2p_3 + p_4)/6$ $p_1 = f(t_k, y_k) \cdot h$ $p_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{p_1}{2}\right) \cdot h$ $p_3 = f\left(t_k + \frac{h}{2}, y_k + \frac{p_2}{2}\right) \cdot h$ $p_4 = f(t_k + h, y_k + p_3) \cdot h$
Ikkinchi tartibdagi differensial tenglama uchun $y'' = f(t, y, y')$ $y = f(t_k) = y_k$
$y'_{k+1} = y'_k + \frac{1}{6}(p_1 + 2p_2 + 2p_3 + p_4)$ $y_{k+1} = y_k + y'_k \cdot h + \frac{1}{6}(p_1 + p_2 + p_3) \cdot h$ $p_1 = f(t_k, y_k, y'_k)h$ $p_2 = f\left(t_k + \frac{h}{2}, y_k + y'_k \frac{h}{2}, y'_k + \frac{p_1}{2}\right)h$ $p_3 = f\left(t_k + \frac{h}{2}, y_k + y'_k \frac{h}{2} + \frac{k_1}{4}h, y'_k + \frac{p_2}{2}\right)h$ $p_4 = f\left(t_k + h; y_i + y'_i h + \frac{k_2}{2}h, y'_k + p_3\right)h$

Differensial tenglamalarni sonli yechish uchun yana qator usullar ishlab chiqilgan. Ammo shu mavzuni fan doirasida tushunib olish uchun yuqorida ko'rsatilganlari yetarlidir. Ularni aniq misollarga qo'llab ko'ramiz:

1) Tenglama $y' = t$, boshlang'ich sharti $y(0) = 0$. Topish kerak $y(1) = ?$ Birinchi navbatda, boshlang'ich shartlarini kiritamiz $y(a) = y$, hisoblash $b = 1$ gacha davom etadi. Eyler usuli bilan javobni topamiz:

```

clear
% Boshlang'ich shartlarini kiritamiz.
% Funksiya qiymatlarini massiv ko'rinishida beramiz
a = 0; y(1) = 1; t(1) = a;
% Argumentni chegara qiymatini kiritamiz
b = 1;
% Qadamlar sonini belgilaymiz
k = 10;
% Hisoblash qadamini topamiz

```



```

h = (b-a) / k;
% For sikli yordamida Eyler usulini qo'llaymiz
% Y funksiyaning birinchi qadamdagi qiymat ma'lum bo'lgani
% sababli hisoblashlar n = 2 qadamdan boshlanadi.
% Qdamlar soni n bo'lsa, tugunlar soni n + 1 bo'ladi
for n=2:k+1
    % (28) - formulani kiritamiz
    y(n) = y(n-1) + h * t(n-1);
    t(n) = t(n-1) + h;
    fprintf('y(%3.3f) = %3.3f\n', t(n),y(n));
end
fprintf('Natijada y(%3.5f) = %3.4f\n', t(n),y(n));

```

Bu dasturni ishletsak, quyidagi natija chiqadi:

```

>> misol_16
y(0.100) = 1.000
y(0.200) = 1.010
y(0.300) = 1.030
y(0.400) = 1.060
y(0.500) = 1.100
y(0.600) = 1.150
y(0.700) = 1.210
y(0.800) = 1.280
y(0.900) = 1.360
y(1.000) = 1.450

```

Natijada $y(1.00000) = 1.4500$

Funksiyaning qiymati $y(1) = 1.45$ bo'ldi. Ammo "Differensial tenglamalar" fanidan bu qiymat $y(1) = 1.50$ bo'lishi kerakligi bilamiz. Bunday farq nimadan paydo bo'ldi? Gap shundaki, sonli usullar xar doim yaqinlashgan natijani beradi va u hisoblash qadamining kattaligiga bog'liq. Shu qadam qancha kichik bo'lsa, natija shuncha aniq bo'lib chiqadi. Berilgan misolda $k = 100$, $k=1000$ deb ko'rsangi, natija xaqiqiy qiymatga yaqinlashib kelaveradi.

$k = 100$; %bo'lsa	$k = 1000$; %bo'lsa
Natijada $y(0.00000) = 1.49500$	Natijada $y(0.00000) = 1.49950$

Boshqa misollarni ko'ramiz. Hosilada t argumentidan boshqa o'zgaruvchilari ishtirok etishi mumkin: $y' - t = -y$, $y(0) = 1$, $y(1) = ?$. Bunday turdagi tenglamalarni $y' = f(t, y)$ ko'rinishiga keltiramiz $y' = t - y$. Bu yerda $a = 0$, $b = 1$ va boshlang'ich $y = 1$.

```

a = 0;    b = 1;
y(1) = 1; t(1) = a;
k = 1000;
h = (b-a) / k;
for n=2:k+1
    y(n) = y(n-1) + h * (t(n-1)-y(n-1));
    t(n) = t(n-1) + h;
    %fprintf('y(%3.4f) = %3.4f\n', t(n),y(n));
end
fprintf('Natijada y(%3.5f) = %3.4f\n', t(n),y(n));

```

```

>> misol_16
Natijada y(2.00000) = 0.7354

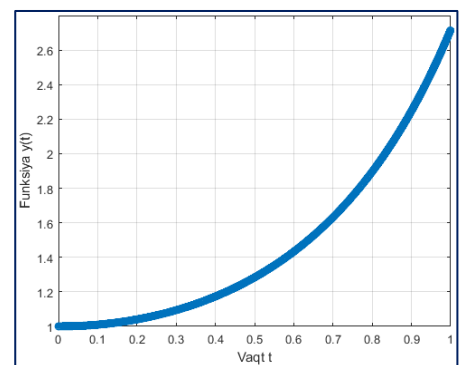
```

Differensial tenglamani sonli yechganda, berilgan hosilani finksiya ko‘rinishida ham bersa bo‘ladi: $y' = 2xy$, $y(0) = 1$, $y(1) = ?$. Shu bilan birga topmoqchi bo‘lgan funksiyamizning vaqt bilan o‘zgarishi grafigini ham chizsak bo‘ladi.

```

a = 0; b = 1; y(1) = 1; t(1) = a;
k = 10000; h = (b-a) / k;
for n=2:k+1
    y(n) = y(n-1) + h * func_01(t(n-1),y(n-1));
    t(n) = t(n-1) + h;
    %fprintf('y(%3.4f) = %3.4f\n', t(n),y(n));
end
fprintf('Natijada y(%3.5f) = %3.4f\n', t(n),y(n));
plot(t,y,'o-'); grid on; xlabel('Vaqt t'); ylabel('Funksiya y(t)');
function yf = func_01(xx,yy)
    yf = 2*xx*yy;
end

```



Ikkinchi tartibdagi differensial tenglamalarni yechganda, sikda yana bitta had qo‘shiladi, xolos. $y'' = xe^{-x}$, $y'(0) = -2$, $y(0) = 3$, $y(1) = ?$

```

a = 0;    b = 1;
t(1) = a;
v(1) = -2; y(1) = 3;

```

```

k = 10000; h = (b-a) / k;
for n=2:k+1
    v(n) = v(n-1) + h * func_01(t(n-1));
    y(n) = y(n-1) + h * v(n-1);
    t(n) = t(n-1) + h;
    fprintf('y(%3.4f) = %3.4f\n', t(n),y(n));
end
fprintf('Natijada y(%3.5f) = %3.4f\n', t(n),y(n));

function yf = func_01(xx)
    yf = xx*exp(-xx);
end

```

Differensial tenglamalarni yechganda, ko‘rsatilgan misollarda qizil rang bilan belgilangan joylarni berilgan masalaga moslashtirib o‘zgartirsangiz yetarlidir.

MATLABda differensial tenglamalarni turli operatorlar yordamida yechsa bo‘ladi: ode45, ode23, ode113, ode23s va boshqalar. Ular Runge-Kutta, Eyler, Adams, Rozenbrok usullariga asoslangan. Tenglama $y' = t^2/2 + y$, $y(0) = 0$, $y(2) = ?$ bo‘lsa, uni quyidagicha yechsa bo‘ladi

```

t_limits = [0 2];
y_init = 0;
[t,y] = ode45(@(t,y) t^2 /2 + y, t_limits, y_init);
y(length(t))
plot(t,y,'bo'); grid on;

```

Bu yerda $t^2/2 + y$ tenglamaning ko‘rinishi, t_limits – vaqt bo‘yicha chegaraviy qiymatlari, y_init – funksiyaning boshlang‘ich qiymati. Ko‘rsatilgan operatorlarni chuqurroq o‘rganish uchun maxsus adabiyotlardan foydalanishni tavsiya qilinadi.

Mexanik harakatlarni modellashtirish

Bu paragrafda mexanikada va umuman fizikada uchraydigan harakatlarni modellashtirib, jismlarning fazodagi harakatini ko‘rib chiqamiz. Shunda mexanika masalalarida ko‘p uchraydigan balandlikdan gorizontol hamda gorizontga burchak ostida otilgan jismlarning harakatlarini misol qilib ko‘ramiz.

Modellashtirishda, masalaning shartiga qarab, ba’zi muhim

shartlarni inobatga olish kerak bo'ladi: dunday harakatlarni ifodalash uchun ikkita x va y koordinatalari talab etiladi. Harakat shu tekislikda yuz beradi. Havoning qarshiligini inobatga olmaymiz, shuning uchun tortishish kuchidan boshqasi ishtirok etmaydi. Harakat Yer sathiga yaqin bo'lgani uchun tezlanishni erkin tushish tezlanishiga teng $a = g$ deb olsa bo'ladi.

Balandlikdan gorizontal otilgan jismning ikkita o'q bo'yicha harakatlanishi tenglamasini quyidagicha yozamiz:

$$\begin{cases} v_x = v_0 \\ v_y = -gt \\ x = v_x t = v_0 t \\ y = y_0 - \frac{gt^2}{2} \end{cases} \quad (3)$$

Bunday harakat uchun boshlang'ich shartlar: $x(0) = 0$ koordinatalar sistemasi boshidan boshlanadi; $y(0) = y_0 = h$ - balandlik; $x'(0) = v_0$; $y'(0) = 0$. Hizsoblashlarni jism Yer sathiga yetganda, ya'ni uning joriy balandligi $y = 0$ bo'lganda to'xtatamiz. Shu harakatni aks etadigan dastur, izohlari bilan birga, quyida keltirilgan:

```
clear; close all;
% Jismning boshlang'ich koordinatalari va tezligi
x_start = 0; y_start = 90; v_start = 10;
% Hisoblashni tugatish sharti
y_end = 0;
t = 0; dt = 0.01; % Boshlang'ich vaqt va hisoblash intervali

figure; % Grafik yangi oynada ochilishini ko'rsatamiz

yline(1,'k-','LineWidth',5); % Vertikal chizig'i belgilanadi
xline(1,'b-','LineWidth',5); % Gorizontal chiziq bilan Yer sathini
ko'rsatamiz

axis([-60 60 -10 100]); % Grafikning chegaralari belgilanadi
hold on; grid on; % Yangi nuqtalarni qo'shiladi hamda setka kiritildi

xlabel('Yer sathi'); ylabel('Balandlik'); ax = gca; ax.FontSize = 12; %
Yozuvlar
```

```

y = y_start; % Vertikal o'qi bo'yicha o'zgarishi kiritildi
k=0; % Hisoblash indeksi
g = 10; % Erkin tushish tezlanishi
m = 1; % Jismning massasini ham kiritsa bo'ladi

% Hisoblashni to'xtash shartini kiritamiz,
% jism Yer sathiga yetganda harakat to'xtaydi
while y >= y_end %shunda hisoblash ham to'xtatiladi
    k = k + 1;
    % har qadamda yangi koordinatlar, tezliklar va vaqt hisoblanadi
    x_new(k) = v_start * t; y_new(k) = y_start - g * t^2/2; t_new(k) =
t;
    vx_new(k) = v_start; vy_new(k) = g*t;
    % Yangi koordinatalarni grafikda ko'rsatiladi
    plot(x_new(k),y_new(k),'o','MarkerSize',6,'MarkerFaceColor','g')

    % Shu siklda jismning tezligi va energiyasini topsa bo'ladi
    V_new(k) = sqrt(vx_new(k)^2 + vy_new(k)^2);
    P_new(k) = y_new(k) * m * g; K_new(k) = m * V_new(k)^2 * 0.5;
    E_new(k) = K_new(k) + P_new(k);

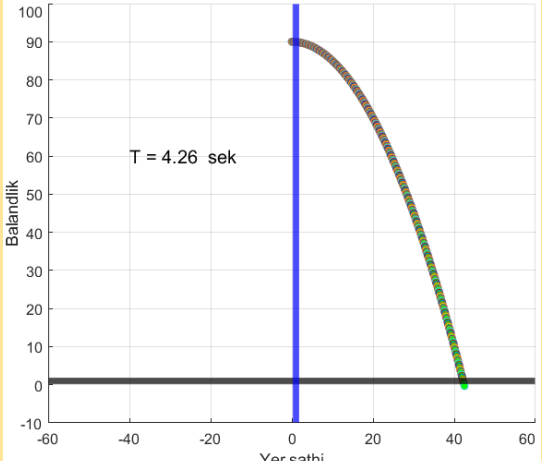
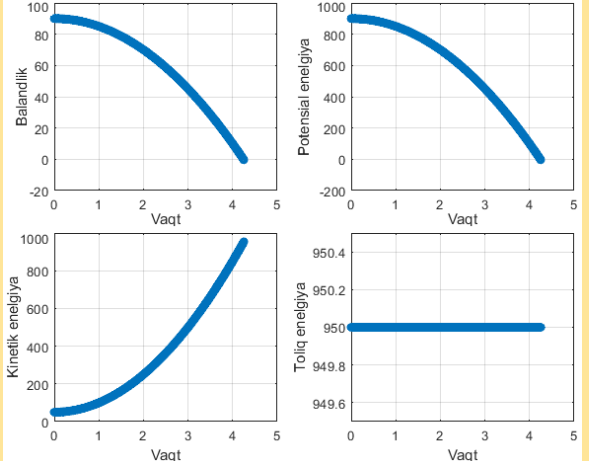
    t = t + dt; % Vaqt o'zgardi
    y = y_new(k); % Balandlik belgisi o'zgarishi ko'rsatiladi

    pause(0.05); % Keyingi siklga vaqtni kechikib (sekundda) o'tsa bo'ladi
end
% Natijalar Buyruqlar Oynasida ko'rsatiladi
fprintf('\n T = %2.4f sekund, X = %2.4f metr, Y = %2.4f metr \n', t,
x_new(k), y_new(k));
% Tushish vaqtni grafikda ko'rsatsa bo'ladi
text(-40,60, 'T = ' + string(t) + ' sek','FontSize', 15)

```

Asosiy harakatlanishdan tashqari, harakatni tavsiflovchi boshqa kattaliklarning o'zgarishlarini ham ko'rsa bo'ladi. Buning uchun grafiklarga yangi oyna ochib, istalgan kattaliklarni tahlil qilish mumkin. Masalan:

figure subplot(2,2,1) plot(t_new, y_new,'o','MarkerSize',6) grid on	subplot(2,2,3) plot(t_new, K_new,'o','MarkerSize',6) grid on
--	--

<pre> xlabel('Vaqt'); ylabel('Balandlik'); ax =gca; ax.FontSize = 10; </pre>	<pre> xlabel('Vaqt'); ylabel('Kinetik enelgiya'); ax =gca; ax.FontSize = 10; </pre>
<pre> subplot(2,2,2) plot(t_new, P_new,'o','MarkerSize',6) grid on xlabel('Vaqt'); ylabel('Potensial enelgiya'); ax =gca; ax.FontSize = 10; </pre>	<pre> subplot(2,2,4) ylim([E_new(k) - 0.5 E_new(k) + 0.5]) plot(t_new, E_new,'o','MarkerSize',6) grid on xlabel('Vaqt'); ylabel('Toliq enelgiya'); ax =gca; ax.FontSize = 10; </pre>
	

Natijada ikkita rasm paydo bo‘lib, u yerda ko‘rsatilgan grafiklar chiziladi.

Gorizontga burchak ostida otilgan jismning harakati tenglamasini quyidagicha yozamiz:

$$\begin{cases} v_x = v_0 \cos \alpha \\ v_y = v_0 \sin \alpha - gt \\ x = v_x t = v_0 t \cos \alpha \\ y = v_y t - \frac{gt^2}{2} = v_0 t \sin \alpha - \frac{gt^2}{2} \end{cases} \quad (4)$$

Bunday harakat uchun boshlang‘ich shartlar: $x(0) = y(0) = 0$ koordinatalar sistemasi boshidan boshlanadi; $x'(0) = v_0 \cos \alpha$; $y'(0) = v_0 \sin \alpha$. Hisoblashlarni jism Yer sathiga qayta yetganda, ya’ni uning joriy balandligi ikkinchi marta $y = 0$ bo‘lganda to‘xtatiladi:

```

clear; close all;
x_start = 0; y_start = 1; % Jismning boshlang‘ich vaziyatlari belgilanadi
v_start = 45; alpha = 30; % Jismning boshlang‘ich tezligi va burchagi
y_end = -0.01; % Hisoblashni tugatish sharti
t = 0; dt = 0.01; % Boshlang‘ich vaqt va hisoblash intervali

```

```

figure ; % Grafik yangi oynada ochiladi
yline(1,'k-','LineWidth',5); xline(1,'b-','LineWidth',5);
axis([-100 100 -10 50]); % Grafikning chegaralari belgilanadi
hold on; grid on;
xlabel('Yer sathi'); ylabel('Balandlik');
ax =gca; ax.FontSize = 12;
y = y_start; k=0; % Hisoblash indeksi
g = 10; m = 1; % Erkin tushish tezlanishi va jismning massasini ham
kiritsa bo'ladi
while y >= y_end
    k = k + 1;
    % Har qadamda yangi koordinatlar va vaqt hisoblanadi
    vx_new(k) = v_start * cosd(alpha);  vy_new(k) = v_start * sind(alpha)
- g * t;
    v_new(k) = sqrt(vx_new(k)^2 + vy_new(k)^2);
    x_new(k) = vx_new(k) * t;  y_new(k) = vy_new(k) * t;
    t_new(k) = t;
    plot(x_new(k),y_new(k),'o','MarkerSize',6,'MarkerFaceColor','g')
    P_new(k) = y_new(k) * m * g;  K_new(k) = m * v_new(k)^2 * 0.5;
    E_new(k) = K_new(k) + P_new(k);
    t = t + dt; % Vaqt o'zgardi
    y = y_new(k); % Balandlik belgisi o'zgarishini ko'rsatdik
    pause(0.02);
end
fprintf('\n T = %2.4f sekund, X = %2.4f metr, Y = %2.4f metr \n', t,
x_new(k), y_new(k));

```

Yuqoridagi misolga o'xshab, bu yerda ham grafiklarda boshqa istalgan kattalikning o'zgarishini ko'rsa bo'ladi. Buning uchun qo'shimcha buyruqlarni kiritish kerak.

NAZORAT SAVOLLARI

1. Fizika va astronomiyada integrallashning mohiyati qanaqa?
2. Integrallash ta'rifini keltiring
3. Integrallashning fizik ma'nosi qanaqa?
4. Sonli integrallashning qanaqa usullari bor?
5. Integrallash va differensial tenglamani yechishda aniqlik qaysi kattalikga bog'liq?
6. Simpson usulini qo'llash kerak bo'lsa, qaysi kutubxonaga burojaat qilinadi?

7. Cheksizlikni qaday belgilasa bo'ladi?
8. Oddiy differensial tenglamaning tartibi nima bilan belgilanadi?
9. Differensial tenglamalarni sonli yechish usullarining qaysilarin bilasiz?
10. Fizik jarayonlarga tegishli differensial tenglamalarning qanday misollari bor?

AMALIY MASHG'ULOTLAR MAZMUNI

1-AMALIY MASHG'ULOT. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari

Reja:

1. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari.
2. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish.

Bu yerda oddiy matematik va fizik masalalarni yechish kerak bo'ladi. O'zizni tekshirish uchun qavslarda yechimlar ham berilgan. Masalani yechish uchun biron bir formula kerak bo'lib qolsa, uni o'zingiz topishiz kerak bo'ladi. Spyder dasturnida algatirimlang.

1	a va b nomli ikkita haqiqiy son berilgan. Ularning yig'indisi, ayirmasi, ko'paytmasi hamda nisbatni toping.
2	$\frac{ a - b }{1 + ab }$ hisoblang. ($a = 5$, $b = -9$ javob $= -0.0870$; $a = 0.5$, $b = -0.9$ javob $= -0.2759$)
3	Kubning tomoni berilgan. Uning hajmini va tomonlarining yuzini toping ($a = 5$, javob: hajm $= 125$, tomonlar yuzasi $= 150$)
4	Toshning h metr balandlikdan tushish vaqtini toping. ($g=10$, $h = 10$ bo'lsa $t = 1.41$; $h = 500$ bo'lsa $t = 10$)
5	$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ y }}{1 + \frac{x^2}{3}}$

	$(x = 3, y = -5, a = -0.0739; x = -1.3, y = 0.45, a = 0.4799)$
6	$b = x(\arctan z + e^{-(x+1)})$ $(x = 2, z = 0.4, b = 0.8606; x = 0.2, z = -0.4, b = -0.0159)$
7	$a = \frac{3 + e^{y-1}}{1 + x^2 y - \tan z }$ $(x = 2, y = 0.4, z = 0.1, a = 1.6141; x = -0.6, y = 1.2, z = -0.2, a = 2.8050)$
8	$b = 1 + y - x + \frac{(y - x)^2}{2} + \frac{(y - x)^3}{3}$ $(x = 0.2, y = -1.2, b = 2.4653; x = -1.2, y = 0.2, b = 4.2947)$
9	$a = (1 + y) \frac{x + y/(x^2 + 4)}{e^{-x-2} + 1/(x^2 + 4)}$ $(x = -1.2, y = 0.2, a = -2.2047; x = 0.2, y = -1.2, a = 0.0542)$
10	$b = \frac{1 + \cos(y - 2)}{x^2/3 + \cos z}$ $(x = -1.2, y = 0.2, z = 1.1, b = 0.8278; x = 1.2, y = 2.2, z = -0.1, b = 1.3424)$
11	$a = y + \frac{x}{y^2 + \left \frac{x^2}{y+2} \right }$ $(x = 1.2, y = 2.2, a = 2.4315; x = -0.2, y = 0.2, a = -3.2375)$
12	$b = (1 + \tan^2 z/2)$ $(z = 1.2, b = 4.3080; z = 2.2; b = 1.9437; z = -0.2; b = 1.0205)$
13	$a = \frac{2 \cos(x - \pi/6)}{0.5 + \sin^2 y}$ $(x = -0.2, y = 0.2, a = 2.7784; x = -1.2, y = 0.2, a = -0.5643)$
14	$b = 1 + \frac{z^2}{3 + z/5}$ $(z = 1.2, b = 1.4444; z = -1.2, b = 1.5217; z = -2.2; b = 2.8906)$
15	Kvadratning tomoni berilgan. Uning yuzini va perimetrini toping.
16	To'g'ri burchakli uchburchakning katetlari berilgan. Uning gipotenuzasi, perimetri va yuzini toping

Bu yerda oddiy fizik masalalar yechimini oddiy chiziqli algoritmlar asosida tuzish kerak bo‘ladi. Masalani yechish uchun biron bir formula kerak bo‘lib qolsa, uni o‘zingiz topishingiz kerak bo‘ladi.

1	t_1 haroratli suvning hajmi v_1 . U haroratu t_2 бўлган v_2 hajmdagi suv bilan aralashtirildi. Natijada, hosil bo‘lgan aralashmaning hajmini va haroratini toping.
2	Uchta qarshilik R_1, R_2, R_3 bir-biriga parallel ulangan. Ularning umumiy qarshiligini toping.
3	Uchta qarshilik R_1, R_2, R_3 bir-biriga ketma-ket ulangan. Ularning umumiy qarshiligini toping.
4	Bir-biridan r masofada joylashgan m_1 va m_2 massalar orasidagi F tortishish kuchini toping.
5	Koordinatalari x_1, y_1 va x_2, y_2 bo‘lgan ikkita nuqta orasidagi masofani toping.
6	Uchburchak uchlari koordinatalar bilan berilgan. A) uchburchakning perimetrini va B) yuzini hisoblab topish kerak.
7	x haqiqiy son berilgan. Faqat ko‘paytirish, qo‘shish, ayirish va bo‘lish operatorlardan foydalanib, quyidagilarni hisoblang $1 - 2x + 3x^2 - 4x^3$ $1 + 2x + 3x^2 + 5x^3$
8	Jism yo‘lning birinchi yarimini $t_1 = 2$ sek davomida, ikkinchisini $t_2 = 8$ sek davomida bosib o‘tdi. Agar yo‘lning uzunligi $s = 20$ metr bo‘lsa, o‘rtacha tezli $\langle v \rangle$ aniqlansin.
9	Tosh $h = 1200$ metr balandlikdan tushmoqda. Tosh tushish vaqtining so‘ngi sekundida qanday s yo‘lni o‘tadi?
10	Yuqoriga tik otilgan jism $h = 8.6$ metr balandlikda $\Delta t = 3$ sek oraliq bilan ikki marta bo‘ldi. Havoning qarshiligini hisobga olmay, otilgan toshning boshlang‘ich tezligini hisoblang.
11	Nuqta $R = 10$ metr radiusli aylana yoyi bo‘ylab harakatlanmoqda. Vaqtning biror momentida nuqtaning normal tezlanishi $a_n = 4.9$ metr/sek ² ; shu momentda to‘la va normal tezlanish vektorlari $\varphi = 60^\circ$ burchakni tashkil qiladi. Nuqtaning tezligi v va tangensial tezlanishi a_t topilsin.
12	Minoradan gorizontaal yo‘nalishda tosh otdilar. Tosh $t = 2$ sek dan keyin minora asosidan $S = 40$ metr masofada yerga tushdi. Toshning boshlang‘ich v_0 va oxirgi v tezliklari aniqlansin

13	Yer sirtidan tik yuqoriga qarab uchirilgan $m = 1$ tonnaga teng massali fazoviy kema $a = 2g$ tezlanish bilan ko'tarilmoqda. Soplodan otilib chiqayotgan gaz oqimining tezligi $v = 1200 \text{ m/s}$. Sarflanayotgan yoqilg'i miqdori Q_m topilsin.
14	Ikkita $m_1 = 10 \text{ kg}$ va $m_2 = 15 \text{ kg}$ massali yuklar bir-biriga tegadigan qilib, uzunligi $L = 2$ metr ipga osilgan. Kichik yuk $\varphi = 60^\circ$ burchakka og'dirib qo'yib yuborildi. Yuklarning urilishini noquyishqoq deb hisoblab, urilishdan so'ng ikkala yukning ko'tarilish balandligi h aniqlansin.
15	Marsning yulduz davri 1.88 Yer yili bo'lsa, uning sinodik davri qancha?
16	Qo'shaloq yulduz α – Cen aylanish davri 79 yil . Orbitasining katta yarim o'qi $17.6''$, yillik parallaks yesa $0.75''$ tashkil etadi. Yulduz komponentlarining massalarini va massalari yig'indisini alohida aniqlang (ular massa markazidan 3:4 masofalarga ajratilgan deb oling).
17	Saturnning massasini aniqlang (Yer massasi birligida). Saturn yo'ldoshi Titan undan 1220 ming km masofada joylashgan va 16 kunlik davr bilan uning atrofida aylanadi.
18	Yangi yulduz yorqinlanishi vaqtida ravshanligi 3.2^m yulduz kattaligiga yetadi. Bu turdagi yangi yulduzlar -8^m absolyut kattalikka ega ekanligi ma'lum bo'lsa, unga bo'lgan masofani hisoblang.
19	Elektron $v = 0.6 \cdot c$ tezlik bilan harakatlanmoqda. Elektronning relyativistik impulsi r aniqlansin.

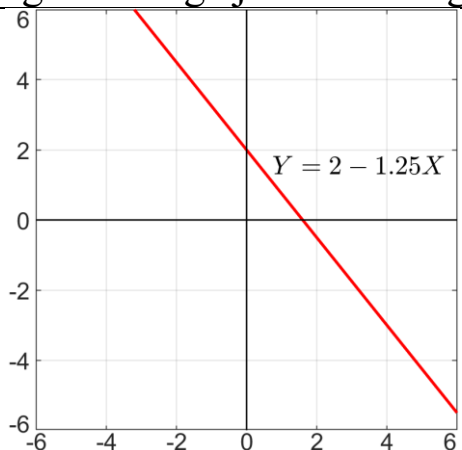
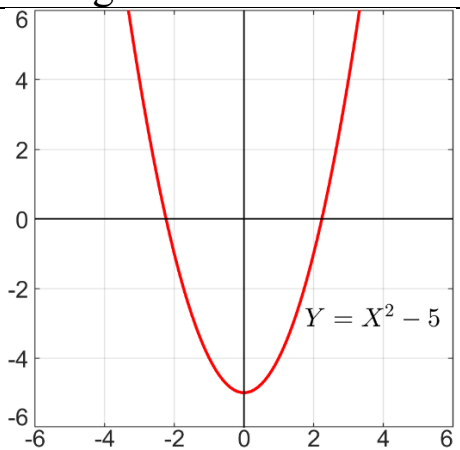
2-AMALIY MASHG'ULOT: Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari

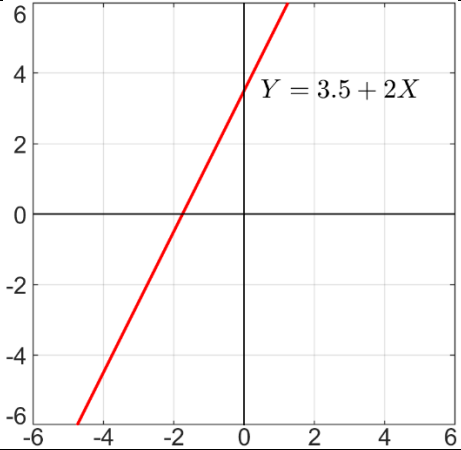
Reja

1. Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari.
2. Chala tarmoqli va to'liq tarmoqli algoritmlar.
3. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash.
4. Mantiqiy operatsiyalar.
5. Sikllar bilan ishlash: sikllar va siklik operatorlar.
6. Sikllar yordamida summa va ko'paytmalarni topish.
7. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.

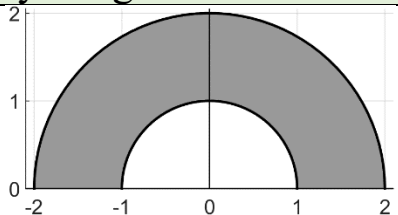
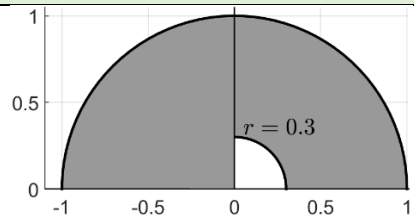
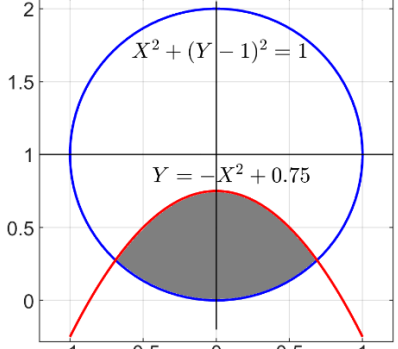
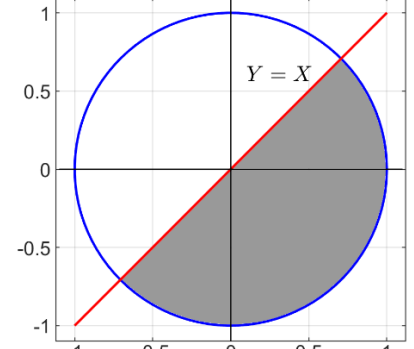
Bu yerda munosabatlar va shartli operatorlar yordamida sodda mantiqiy masalalarni yeching.

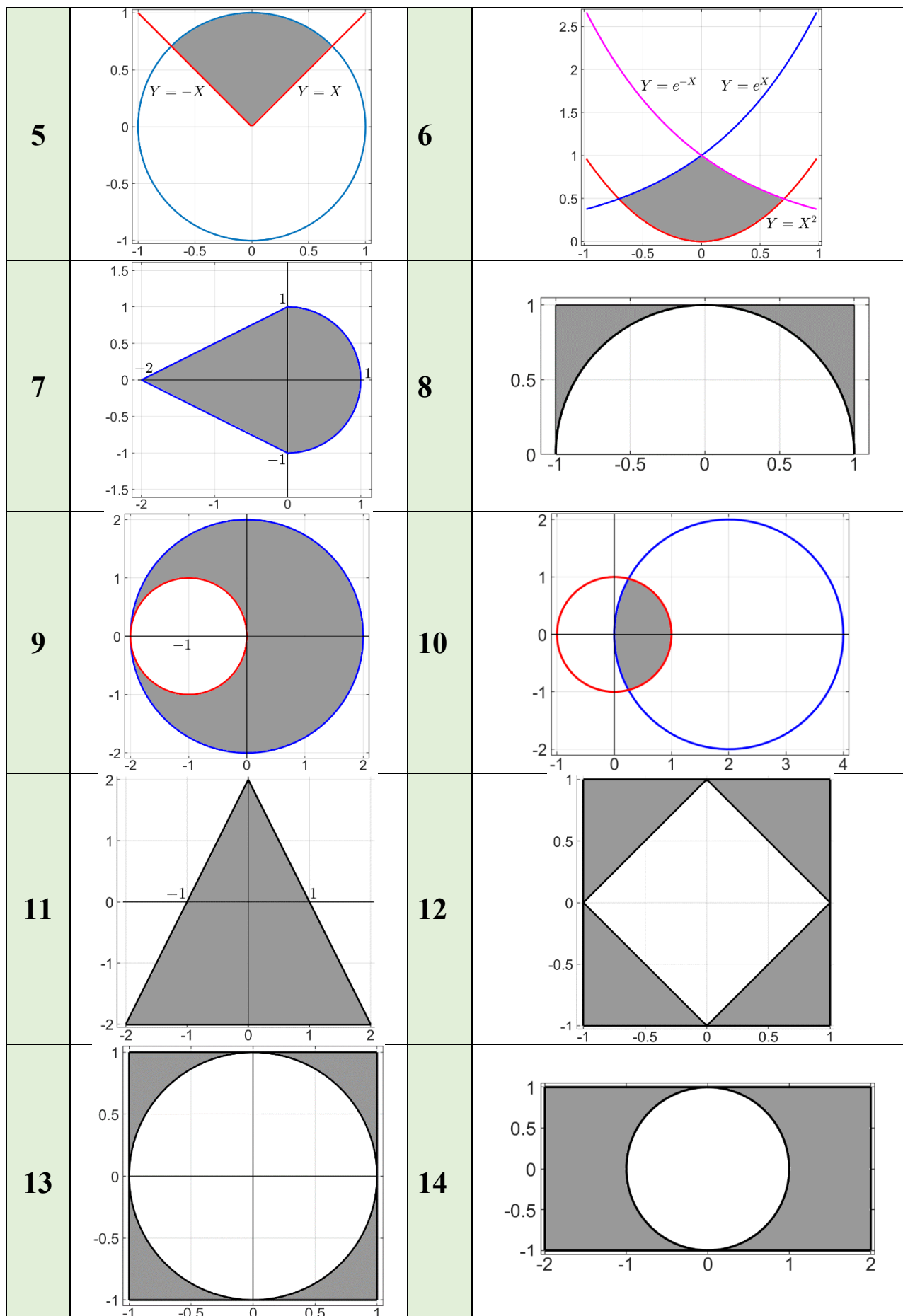
1	$z + a = 0$ ko'rinishdagi tenglamaning yechimini dasturlang (a – ma'lum deb oling)
2	$ax^2 + bx + c = 0$ ko'rinishdagi kvadrat tenglama yechimini beradigan dastruni tuzing. Diskriminantning nolga nisbatan hamma holatlarini inobatga oling.
3	$ax^3 + bx^2 + cx + d = 0$ ko'rinishdagi kubik tenglama yechimini beradigan dasturni tuzing. Diskriminantning nolga nisbatan hamma holatlarini inobatga oling.
4	x va y sonlar berilgan bo'lsin. Ular orasida eng kattasini – $\max(x, y)$ va eng kichigini – $\min(x, y)$ toping.
5	x, y va z sonlar berilgan bo'lsin. Ular orasida eng kichigini – $m \in (x, y, z)$ toping.
6	x, y va z sonlar berilgan bo'lsin. Ular orasida eng kattasini – $\max(x, y, z)$ toping.
7	x, y va z sonlar berilgan bo'lsin. Ular kombinatsilari orasida – $\max(x + y + z, xyz)$ toping.
8	x, y va z sonlar berilgan bo'lsin. Ular kombinatsilari orasida – $m \in \min^2\left(x + y + \frac{z}{2}, xyz\right) - 1$ toping.
9	Mantiqiy ifodaning $x^2 + y^2 \leq 4 (x = 1, y = -1)$ qiymatini toping.
10	Mantiqiy ifodaning $x^2 - y^2 \leq 3 (x = -1, y = 3)$ qiymatini toping.
11	Mantiqiy ifodaning $x * y > 0 (x = 1, y = -2)$ qiymatini toping.

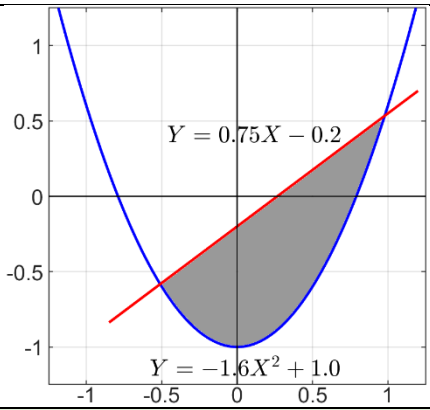
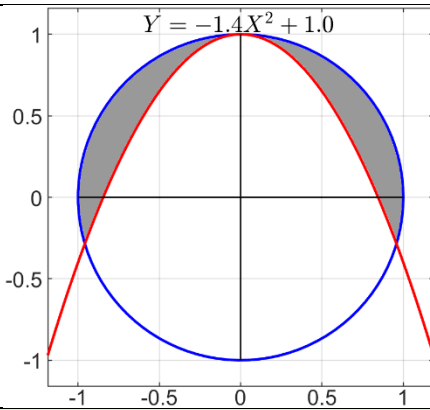
12	Mantiqiy ifodaning $x * y \neq 0(x = -1, y = -2)$ qiymatini toping.		
13	Ixtiyoriy butun sonning juft yoki toqligni aniqlang, 3 songa karra ekanmi, yoki yo‘qligini aniqlab bering.		
14	Ixtiyoriy butun son 3 soniga karra ekanmi, yoki yo‘qligini aniqlab bering.		
15	Ixtiyoriy butun son 5 songia karra ekanmi, yoki yo‘qligini aniqlab bering.		
16	Ixtiyoriy son 3 xonalimi yoki yo‘qligini aniqlang		
17	Ixtiyoriy son 4 xonalimi yoki yo‘qligini aniqlang		
18	x va y sonlar sinov nuqtasining dekart koordinatalari bo‘lsin. Shu nuqta radiusi r ixtiyoriy bo‘lgan va markazi koordinatalar sistemasi boshida yotgan aylana ichida yoki tasqarisidaligini aniqlaydigan algoritm va dastur tuzing.		
19	x va y sonlar sinov nuqtasining dekart koordinatalari bo‘lsin. Shu nuqta radiusi r ixtiyoriy bo‘lgan va markazi ixtiyoriy joyda yotgan aylana ichida yoki tasqarisidaligini aniqlaydigan algoritm va dastur tuzing.		
20	Katta va kichik yarimo‘qlari a va b bo‘lgan ellipsoid bo‘lsin. (x, y) koordinatali sinov nuqtasi shu shakl ichidami yoki yo‘qmi degan savolga javob beradigan dastur tuzing.		
21		22	

23		Sinov nuqtaning dekart sistemasidagi koordinatalari (x, y) bo'lsin. Shu nuqta berilgan chiziqlarga nisbatan yuqori yoki quyi yarim tekisligida joylashganligini aniqlang.
24	switch ... case mantiqiy operatori yordamida oyning raqamiga qarab, nomini beradigan dastruni tuzing.	
25	Ilgari soliqlar stavkasi oylik maoshning miqdoriga qarab o'zgarar edi. Shunda eng kichik ish haqqidan soliq olinmasdi, 1 baravaridan 5 baravarigacha soliq stavkasi 7.5% tashkil qilar edi. 5 baravaridan 10 baravarigacha – 16.5%, undan ko'pidan 22.5% soliq olinardi. Ixtiyoriy oylik maoshdan kelib chiqib, soliq stavkasini hisoblaydigan dastruni tuzing. Eng kichik ish haqini ommabop ma'lumot manbalaridan bilib olishingiz mumkin.	

2-Topshiriq. Bu yerda mantiqiy operatsiyalar bilan bog'liq bo'lgan masalalarni yechish kerak bo'ladi.

Sinov nuqtasi bo'yalgan sohaga tegishliligini aniqlaydigan mantiqiy ifodani yozing va dasturni tuzing			
1		2	
3		4	



15		16	
Masalalarni yeching. x, a, b kabi kattaliklarni o'zingiz tanlab kiriting			
17	$y = \begin{cases} \cot(x^2 e^{3k}) + \ln r + x & \text{agar } x = rs \\ \sqrt[5]{x^2} + \sqrt{ \arcsin k } & \text{agar } x > rs \\ \arctan(kx + \tan(rs)) & \text{agar } x < rs \end{cases}$		
18	$y = \begin{cases} \sin e^{a+b} + x^2 & \text{agar } a + b > x \\ \arctan(abc) + \sqrt[3]{x} & \text{agar } a + b = x \\ \arcsin(\cos^2(\sqrt{ x })) & \text{agar } a + b < x \end{cases}$		
19	$y = \begin{cases} 0 & \text{agar } x \leq 0 \\ x^2 - x & \text{agar } 0 < x \leq 1 \\ x^2 - \sin \pi x^2 & \text{boshqa hollarda} \end{cases}$		
20	$y = \begin{cases} abx - (\cos zx)^2 & \text{agar } x \leq 3,5a \\ (a - x)^2 - \ln(z + x) & \text{agar } 3.5a \leq x \leq b \\ \sqrt{bx - a + zx^2} & \text{agar } x > b \end{cases}$		
21	$y = \begin{cases} \ln mx + n & \text{agar } x^2 > m + n \\ e^{\cos mx-n } & \text{agar } x^2 = m + n \\ \sqrt[3]{k^2 + \cos^2 x} & \text{agar } x^2 < m + n \end{cases}$		
22	$y = \begin{cases} \sqrt{ ax - \cos^2(b^5 x) + 5.1c^2 } & \text{agar } 1 - x^2 = a + c \\ e^{0.04x} + \ln b^5 \cos x & \text{agar } 1 - x^2 > a + c \\ \cos^2(b^3 x^2) + \ln bx - a^2 & \text{agar } 1 - x^2 < a + c \end{cases}$		
23	$y = \begin{cases} \sqrt[3]{b^2 + \sqrt{ x + c }} & \text{agar } \lg a < x \\ \cos(x - b - c) & \text{agar } \lg a = x \\ \sin(x + a - b) & \text{agar } \lg a > x \end{cases}$		

24	$y = \begin{cases} (3.5a - 7.3bx + \sin(zx))^3 & \text{agar } x < \ln a \\ a^b - \cos^3(a + zx) & \text{agar } -\ln a < x < b \\ \sqrt{ \tan a - x } - x^2 & \text{agar } x > b \end{cases}$		
Ko'rsatilgan shartlar bajarilganda dastur HAQIQAT (TRUE) so'zini qaytarishi kerak.			
25	A va B sonlarning faqat bittasi juft bo'lsa.	26	$x > 3$ yoki $x < -1$ bo'lsa
27	$-4 < y \leq 2$ va $x > 0$ bo'lsa	28	$0 < y \leq 6$ va $x < 7$ bo'lsa
29	Butun N to'rtga yoki yettiga karra bo'lsa	30	Berilgan butun N 7 bo'linib, 11 va 13 ga qoldiqsiz bo'linmasa
31	To'rtta A,B,C,D sonlar ichida bittasi 3 ga karrali, va yana bittasi 5 ga karralili bo'lsa	32	Berilgan butun N 3 bo'linadi, ammo 9 ga qoldiqsiz bo'linmaydi, agar u 4 ta bo'linsa, 5 va 24 ga qoldiqsiz bo'linsa
33	Uchta A, B, C sonlarning hammasi 3 ga karali bo'lsa	34	To'rtta A, B, C ,D sonlar ichida ikkitasi juft son bo'lsa
35	Berilgan A -10 dan -1 gacha intervalga yoki 2 dan 15 gacha intervaliga tegishli bo'lsa.	36	Berilgan butun N 3 bo'linmaydi, ammo 7 va 10 ga qoldiqsiz bo'linsa
37	Berilgan A aynan 4 xonali sonligini va u 4999 ga teng bo'lmasa	38	Oylik maoshga qo'shimcha haqni hisoblaydigan mantiqiy ifodani yozing. Shunda ish staji 5 yildan 10 yilgacha bo'lsa, qo'shimcha 2% belgilanadi, ish staji 10 yildan 20 yilgacha bo'lsa, qo'shimcha 10% belgilansin.
39	Agarda fizika, matematika va informatika 4 yoki 5 baholarga topshirilgan bo'lsa, talaba stipendiya oladi sharti uchun mantiqiy instruksiyalarni yozing.	40	Depozit bo'yicha foizni hisoblaydigan mantiqiy operatorni yozing: 5,000,000 sumgacha – yillik 20% hisoblanadi, ot 5,000,000 sumdan 10,000,000 sumgacha – yillik 22 % hisoblanadi.

3-Topshiriq. for va while sikllari yordamida ko'rsatilgan summa va ko'paytmalarni toping.

№	1. for operatorini qo'llang va qatorning yig'indisi (yoki ko'paytmasini hisoblaydigan dasturni tuzing) (13-misol). 2. Dasturni ishlab chiqishda nolga yoki cheksizlikga teng bo'ladigan xadlarni hisobdan chiqarish kerakligini inobatga oling. 3. Natijalarni verguldan keyin 3 xona aniqligi bilan ko'rsating 4. Summa yoki ko'paytmalarning yuqori chegaralarini (k, n) , argumentlarni (t, x) ixtiyoriy tanlasangiz bo'ladi.		
1	$Z = \prod_{n=-3}^k \frac{(n+2) n-4 }{(n+3)!}$	2	$Z = \prod_{j=-3}^k \frac{(j+2)j}{j-3} \sum_{i=j}^{k+5} \left(\frac{i+5}{i-11} - 3i \right)$
3	$W = \sum_{i=-2}^k \frac{(-1)^i(i+3)!}{2(i-4)}$	4	$W = \sum_{i=-3}^k \frac{(-1)^i}{(i-5)^2} \prod_{n=i}^{2k} \frac{n^3-8}{n+4}$
5	$U = \prod_{t=2}^k \frac{tx^t}{j-12} \sum_{i=1}^t \left(\frac{i-4}{i-6} \right)$	6	$P = \prod_{j=-1}^k \frac{(j-j^2)k}{j-4} \sum_{i=j}^{k+2} \frac{\sqrt[3]{ i-5 }}{ i-7 }$
7	$S = \sum_{k=1}^n \frac{(-1)^k(k^3-27)}{3(k+2)!}$	8	$S = \sum_{k=1}^n \frac{(-3)^{3k+1}}{2(k-2)x^{3k}} \prod_{m=1}^{k+2} \frac{m^3-8}{m-4}$
9	$Q = \sum_{k=1}^n \frac{(-1)^k(k-7)}{2(n-k)!}$	10	$S = \sum_{k=1}^n \frac{(-2)^{2k+1}}{(k-2)x^{k+1}} \prod_{m=1}^{k+n} \frac{m^2-4}{m-3}$
11	$U = \prod_{t=2}^k \frac{tx^t}{t-3} \sum_{i=j}^t \frac{i-2}{i-7}$	12	$W = \sum_{i=1}^9 \frac{(7-x)^i}{(i-3)^5} \prod_{n=i}^{17} \frac{n^3-8}{n-12}$
13	$W = \sum_{i=-2}^k \frac{(-1)^i(i+3)!}{i-4}$	14	$Z = \prod_{j=-4}^k \frac{k^{(j+2)}j}{j-3} \sum_{i=j}^{k+5} \left(\frac{\sqrt[5]{i+5}}{i-11} \right)$
15	$Q = \sum_{k=1}^n \frac{(-1)^k(x+3)^2}{k!}$	16	$A = \prod_{j=1}^k \frac{(j^2-4)j}{j-k+1} \sum_{i=j}^9 \frac{i-3}{i-7}$

17	$P = \prod_{k=1}^n \left(\frac{k}{k+1} - \cos^k x \right)$	18	$S = \sum_{k=1}^n \frac{(-1)^{3k+1}}{(k-5)x^k} \prod_{m=1}^{k+7} \frac{m^2 - 9}{m - 2}$
19	$P = \prod_{k=1}^n \left(1 + \frac{\sin(kx)}{k!} \right)$	20	$W = \sum_{i=1}^k \frac{(-1)^i}{(i-3)^2} \prod_{n=i}^{2k} \frac{n^3 - 8}{n + 2}$
21	$Q = \sum_{k=0}^n \frac{(-1)^k (k+1)}{k!}$	22	$P = \prod_{j=-1}^k \frac{(j-j^2)j}{j+14} \sum_{i=j}^{k+3} \frac{\sqrt[4]{ i-5 }}{ i-7 }$
23	$Y = \sum_{k=1}^k \frac{(-1)^{2n} (n^2 - 9)^2}{(3n)!}$	24	$Q = \sum_{k=3}^n \frac{(-2)^{k-1}}{(k-5)x^k} \prod_{i=1}^{k+7} \frac{i^3 - 27}{i - 7}$
№	while operatoridan foydalanib, $\varepsilon = 0.001$ aniqligi bilan summalarning qiymatlarini toping.		
25	$\cos x + \frac{\cos(3x)}{9} + \frac{\cos(5x)}{25} + \dots$	26	$1 + \frac{\cos \frac{\pi}{4}}{1!} x + \frac{\cos 2 \frac{\pi}{4}}{2!} x^2 + \dots$
27	$\sin x - \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} - \dots$	28	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$
29	$\cos x + \frac{\cos 2x}{2} + \frac{\cos 3x}{3} + \dots$	30	$x - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$
31	$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots$	32	$x \sin \frac{\pi}{4} + x^2 \sin 2 \frac{\pi}{4} + \dots$
33	$x + \frac{x^5}{5!} + \frac{x^9}{9!} + \frac{x^{13}}{13!} + \dots$	34	$\frac{x \cos \frac{\pi}{3}}{1!} + \frac{x^2 \cos 2 \frac{\pi}{3}}{2!} + \dots$
35	$1 + \frac{\cos(x)}{1!} + \frac{\cos(2x)}{2!} + \dots$	36	$\frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \frac{1}{5} \left(\frac{x-1}{x+1} \right)^5 + \dots$

37	$1 + \frac{3x^2}{1!} + \frac{5x^4}{2!} + \dots$	38	$-\cos x + \frac{\cos 2x}{4} - \frac{\cos 3x}{9} + \dots$
39	$\frac{x^3}{3} - \frac{x^5}{15} + \frac{x^7}{35} - \dots$	40	$\frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \frac{\cos 6x}{35} + \dots$
41	$\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5} + \dots$	42	$1 + \frac{2x}{1!} + \frac{(2x)^2}{2!} + \dots$
43	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots$	44	$1 + \frac{2}{1!}\left(\frac{x}{2}\right) + \frac{5}{2!}\left(\frac{x}{2}\right)^2 + \frac{10}{3!}\left(\frac{x}{2}\right)^3 + \dots$
45	$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$	46	$1 - \frac{2}{2!}x^2 + \frac{5}{4!}x^4 - \frac{10}{6!}x^6 + \dots$
№	while operatoridan foydalanib, $\varepsilon = 0.001$ aniqligi bilan funksiyalar qiymatlarini berilgan argument uchun qiymatlarini toping va MATLABda o'rnatilgan funksiyalar beradigan qiymati bilan solishtiring (12-misol). Topshiriqning shu qismiga tegishli qoshimcha masala va funksiyalarni https://ru.wikipedia.org/wiki/Ряд_Тейлора sahifasidan topib olsa bo'ladi.		
47	$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}, x \in \mathbb{C}$	48	$\ln(1+x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{(n+1)}, -1 < x \leq 1$
49	$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}, x \in \mathbb{C}$	50	$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}, x \in \mathbb{C}$
51	$\arcsin x = \sum_{n=0}^{\infty} \frac{(2n)!}{2^{2n}(n!)^2} \frac{x^2}{2n}, x \leq 1$	52	$\arccos x = \frac{\pi}{2} - \sum_{n=0}^{\infty} \frac{(2n)!}{2^{2n}(n!)^2} \frac{x^2}{2n}, x \leq 1$
53	$\sinh x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}, x \in \mathbb{C}$	54	$\cosh x = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}, x \in \mathbb{C}$

3-AMALIY MASHG'ULOT: Massivlar bilan ishlash. Grafiklar chizish usullari

REJA

4. Bir o'lchamli va ikki o'lchamli massivlarni shakillantirish, elementlari bilan ishlash.
5. Grafiklar chizish usullari.
6. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

1-Topshiriq. Massivlar bilan ishlash. Massivlarni yaratganda tasodifiy sonlar generatoridan foydalansa bo'ladi. Qiymatlarining kattaligi esa qo'yilgan masalaning shartlaridan kelib chiqqan holatda beriladi.

1) Elementlari ixtiyoriy bo'lgan kvadrat matritsani yarating. Maxsus funksiyalardan foydalanmasdan, bu matritsaning a) diagonal elementlarini, b) diagonaldan tepa qismini, c) diagonaldan pastki qismini ekranga chiqaring

2) 12 ta elementdan iborat butun sonli massivni kiriting. Yangi massivda beshinchi elementni birinchining o'rtacha arifmetik qiymatiga almashtiring

3) 11 ta elementdan iborat butun sonli massivni kiriting. Absolyut qiymati o'rtacha kvadratik qiymatidan katta bo'lgan elementlarning sonini aniqlang.

4) 9 ta elementdan iborat butun sonli massivni kiriting. Massivning maksimal va minimal elementlarini joyi bilan almashtiring.

5) 12 ta elementdan iborat butun sonli massivni kiriting. Toq sonlar yoki juft sonlar orasida qaysi biri ko'pligini toping.

6) 15 ta elementdan iborat haqiqiy sonli massivni kiriting. Massivning birinchi elementidan katta bo'lgan elementlarning sonini toping

7) 15 ta elementdan iborat butun sonli massivni kiriting. Yangi massivning elementlari oldingi massivning elementlari va uning o'rtacha arifmetik qiymati orasidagi farqiga teng bo'lsin.

8) 14 ta elementdan iborat butun sonli massivni kiriting. Massivning musbat juft elementlarning sonini va yig'indisini toping.

9) 12 ta elementdan iborat haqiqiy sonli massivni kiriting. Elementlarni kamayish tartibida joylashtiring. Massivning maksimal va minimal elementlarning yig'indisini toping.

10) 15 ta elementdan iborat butun sonli massivni kiriting. Manfiy elementlarning sonini, musbat elementlarning yig'indisini toping.

11) 12 ta elementdan iborat haqiqiy sonli massivni kiriting. Massivning elementlari uchun intervalning chegaralarini aniqlang.

12) 16 ta elementdan iborat butun sonli massivni kiriting. Uchga karrali elementlarni nolga aylantiring. Almashishlarning sonini aniqlang.

13) 12 ta elementdan iborat haqiqiy sonli massivni kiriting. Oxirgi elementdan kichik bo'lgan elementlarning sonini aniqlang.

14) M (12) haqiqiy sonli massiv berilgan. Minimal element va uning indeksining summasini toping.

15) 10 ta elementdan iborat butun sonli ikkita X va Y massivlarni kiriting. Uchinchi S massivni yarating va uning elementlari oldingi massivdagi bir xil elementlardan iborat bo'lsin.

16) 14 ta elementdan iborat butun sonli massivni kiriting. Qiymati juft bo'lgan elementlarning sonini toping.

17) 16 ta elementdan iborat haqiqiy sonli massivni kiriting. Maksimal va minimal elementlarning indekslarini aniqlang.

18) Berilgan butun sonli massivda ikki va undan ko'p marta uchraydigan elementlarni nolga aylantiring.

19) Har biri 12 ta elementdan iborat ikkita butun sonli massivni kiriting. Agarda birinchi massivning elementlari ikkinchisida ham uchrasin, birinchi massivdagi elementlarni nolga aylantiring.

20) Haqiqiy elementlardan iborat ikkita massivni kiriting. Har bir massivda maksimal qiymatlarini toping va ularning massivlardagi o'rnini almashtiring.

21) Butun sonli massiv berilgan. Hamma elementlarning o'rtacha arifmetik qiymatidan katta bo'lgan elementlarning foizdagi ulushini aniqlang.

22) 15 ta elementdan iborat massivni tasodifiy holda quyidagilar bilan to'ldiring: a) $(0...1)$ diapazonda bo'lgan haqiqiy sonlar bilan; b) x haqiqiy sonlari bilan ($22 \leq x < 23$); c) x haqiqiy sonlari bilan ($0 \leq x < 10$); d) x haqiqiy sonlari bilan ($-50 \leq x \leq 50$);

23) 12 ta elementdan iborat haqiqiy sonli massivning qiymatlari $y_i = i^2 - 2i + 9.3 \cos i$ formulaga binoan hisoblangan bo'lsin. Uni ekranga chiqaring. Yangi massivning elementlari oldingining o'rtacha arifmetik qiymatidan kichiklariga teng bo'lsin.

24) Massiv 12 bolaning vazni haqida ma'lumotni saqlashga mo'ljallangan. Tasodifiy sonlar generatori yordamida massivni 50 va 100 oraliqda bo'lgan haqiqiy sonlar bilan to'ldiring.

25) Massiv 12 bolaning bo'yi haqida ma'lumotni saqlashga mo'ljallangan. Tasodifiy sonlar generatori yordamida massivni 163 va 190 oraliqda bo'lgan haqiqiy sonlar bilan to'ldiring.

26) Massiv yuk mashinasiga yuklangan 30 ta buyumning og'irligi haqidagi ma'lumotlarni saqlaydi. Mashinaning yuk hajmi ma'lum. Barcha buyumlarning umumiy og'irligi avtomobilning yuk ko'tarish imkoniyatidan oshmasligini aniqlang.

27) Massivda sentyabr oyining har kuni tushgan yog'ingarchilik miqdori haqidagi ma'lumotlar saqlanadi. Ushbu oyning birinchi, ikkinchi va uchinchi o'n kunliklarida o'rtacha bir kunda qancha yog'ingarchilik tushganini aniqlang.

28) Massivda 20 ta futbol jamoasi qo'lga kiritgan g'alabalar soni haqida ma'lumotlar mavjud. Uchtadan kam g'alaba qozongan jamoalar sonini va nomerlarini aniqlang.

29) Massivda yanvar oyining har kuni tushgan yog'ingarchilik miqdori haqidagi ma'lumotlar saqlanadi. Oyning qaysi kunlarida yog'ingarchilik bo'lmaganligini aniqlang.

30) Guruhdagi har bir kishining vazni ma'lum. Ularning eng og'irining vazni eng yengil vazndan 2 baravar ko'p ekanligi rostmi?

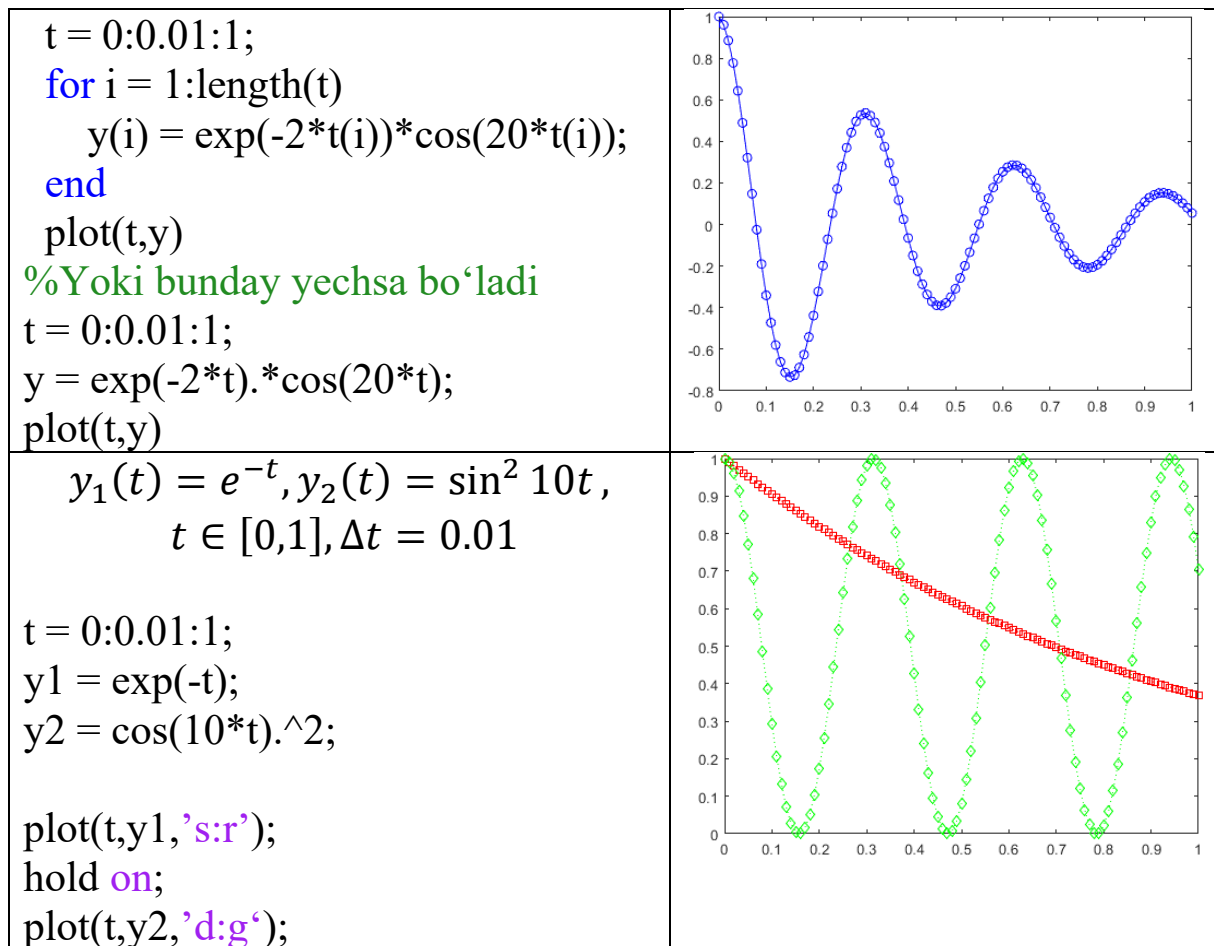
31) Massivdagi qaysi kattalik birinchi bo'lib paydo bo'ladi: maksimal yoki minimal? Agar bunday sonlar bir nechta bo'lsa, unda birinchilarini hisobga olish kerak.

32) 1-Ilovadagi 1-ustunda kvazarlarning kataloglardagi nomi, 2- va 3-ustunlarida ularning ekvatorial koordinatalari berilgan, 4- va 5-ustunlarda kuzatuvlar natijasida topilgan kordinatalar, 5- va 6-ustunlarda esa ularnig ko'rinma yulduz kattaliklari va o'lchash xatoliklari berilgan. Katalogdagi va o'lchangan koordinatalarni bir-biri bilan solishtirib, kvazarlarning qaysi biri qayd etilganini va uning yulduz kattaligi va xatoligini aniqlang. Natijalarni ishchi oynaga chiqaring.

2-Topshiriq. Grafiklarni chizish.

Berilgan funksiyani plot(t,y) operatori yordamida chizing

$$y(t) = e^{-2t} \cdot \cos 20t, \quad t \in [0,1], \Delta t = 0.01$$



- 1) $y(t) = e^{-t} \cdot \sin^3 30t, t \in [0,1], \Delta t = 0.01$
- 2) $y(t) = e^t \cdot \cos 20t, t \in [0,1], \Delta t = 0.01$
- 3) $y(t) = e^t \cdot \sin^2 20t, t \in [0,2], \Delta t = 0.01$
- 4) $y(t) = e^t \cdot \cos^2 20t, t \in [0,2], \Delta t = 0.01$
- 5) $y(t) = \sqrt{t} \cdot \sin^2 10t, t \in [0,4], \Delta t = 0.02$
- 6) $y(t) = t \cdot \sin^2 10t, t \in [0,2], \Delta t = 0.01$
- 7) $y(t) = (1 - t) \cdot \cos 20t, t \in [0,2], \Delta t = 0.01$
- 8) $y(t) = (1 - e^{-t}) \cdot \cos 20t, t \in [0,2], \Delta t = 0.01$
- 9) $y_1(t) = 10e^t, y_2(t) = \sin^3 20t, t \in [0,2], \Delta t = 0.01$
- 10) $y_1(t) = 10\sqrt{t}, y_2(t) = \sin 10t, t \in [0,4], \Delta t = 0.02$
- 11) $y_1(t) = 10(1 - t), y_2(t) = \cos 20t, t \in [0,2], \Delta t = 0.01$
- 12) $y_1(t) = 10(1 - e^t), y_2(t) = \cos 20t, t \in [0,2], \Delta t = 0.01$
- 13) $y_1(t) = 0.1 \cdot e^{-t}, t \in [0,2\pi], \Delta t = 0.1$, Sariq, kvadrat, shtrix/punktir
 $y_2(t) = 60 \sin^2 t, t \in [0,2\pi], \Delta t = 0.1$, Pushti, romb, shtrix
- 14) $y_1(t) = 0.2 \cdot e^{-t}, t \in [0,2\pi], \Delta t = 0.1$, Moviy, plyus, uzliksiz
 $y_2(t) = 40 \cos^2 2t, t \in [0,2\pi], \Delta t = 0.1$, Qizil, xoch, punktir
- 15) $y_1(t) = 0.3 \cdot e^{-2t}, t \in [0,2\pi], \Delta t = 0.2$, Binafsha, romb, punktir
 $y_2(t) = 30 \sin^3 3.5t, t \in [0,2\pi], \Delta t = 0.2$, Yashil, aylana, shtrix

- 16) $y_1(t) = 1 - e^{-t}, t \in [0, 2\pi], \Delta t = 0.2$, Ko'k, plyus, shtrix
 $y_2(t) = 3 \cos 2t, t \in [0, 2\pi], \Delta t = 0.2$, Qora, kvadrat, uzliksiz
- 17) erf(x) va errorbar(x,y,eb) operatorlari yordamida xatolik bilan grafiklarni chizing
- 18) $y(t) = 4 \cdot \arctan t, t \in [0, 5], \Delta t = 0.2$, Qizil, romb, punktir
- 19) $y(t) = 3 \cdot \operatorname{arccot} t, t \in [0, 1.5], \Delta t = 0.3$, Ko'k, kvadrat, shtrix
- 20) $y(t) = 2 \cdot \operatorname{asin}|t|, t \in [-1, 1], \Delta t = 0.1$, Yashil, plyus, shtrix/
punktir
- 21) $y(t) = 3 \cdot \operatorname{acos}|t|, t \in [-1, 1], \Delta t = 0.1$, Binafsha, yulduz, uzliksiz
- 22) $y(t) = t^{2/3}, t \in [0, 10], \Delta t = 0.25$, Moviy, uchburchak, punktir
- 23) $y(t) = t^{-2}, t \in [0.5, 2], \Delta t = 0.05$, Qora, aylana, shtrix
- 24) sdss2124_mags, sdss0806_mags va boshqa fayllardan ravshanlik egri ichiziqlarini o'qib olib, ravshanlik egri chiziqlarini chizing. Chiziq va nuqtalarning ranglari, shakllarini ixtiyoriy tanlab oling.
- 25) 3-o'lchamli sferik funksiyalarni plot3(x,y,z), comet3(x,y,z) operatorlar yordamida chizib ko'ring

- 26) Sferik spillar:

$$x = \sin \frac{t}{2c} \cdot \cos t; y = \sin \frac{t}{2c} \cdot \sin t; z = \cos \frac{t}{2c}; c = 5; 0 < t < 20\pi$$

- 27) Fazodagi sinus funksiyasi:

$$x = \cos t \cdot \sqrt{b^2 - c^2 \cos^2 at}; y = \sin t \cdot \sqrt{b^2 - c^2 \cos^2 at}; \\ z = c \cdot \cos at; a = 10; b = 1; c = 0.3; 0 < t < 2\pi$$

- 28) 3-o'lchamli sferik funksiyalarni surf operatori yordamida chizib ko'ring

- 29) Sinusning sirti $x = \sin u; y = \sin v; z = \sin u + v;$
 $0 < u < 2\pi; 0 < v < 2\pi$

- 30) Elliptik tor (xalqa)

$$x = [1 - r_1 \cos v] \cos u; y = [1 - r_1 \cos v] \sin u; z = r_2 \cdot \left[\sin u + \frac{tu}{\pi} \right]; \\ t = 1.5; r_1 = r_2 = 0.5; 0 < u < 2\pi; 0 < v < 2\pi$$

4-AMALIY MASHG'ULOT Sonli integrallash usullari. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmalri

Reja:

1. Sonli integrallash usullari va algritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.
2. Aniq integrallarni hisoblash.
3. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmalri

1-Topshiriq. Integrallarni hisoblang.

$\int_0^{1.2} \frac{dx}{\sqrt[3]{x+1}}$	$\int_0^3 \sqrt{4+x^2} dx$	$\int_0^2 \frac{dx}{\sqrt{1+x^4}}$	$\int_0^2 e^{-x} \cos \frac{\pi x}{4} dx$
$\int_0^{\pi/6} \sqrt{\tan x} dx$	$\int_{\pi/2}^{\pi} \frac{\cos x}{x} dx$	$\int_1^5 \frac{e^x}{x} dx$	$\int_0^5 (x - 5)^2 x dx$
$\int_0^{\pi} \sqrt{1 + \cos^2 x} dx$	$\int_0^1 e^{2x} \sin(2x^2 + 1) dx$		
$\int_{0.2}^{0.56} \arccos e^{-\sqrt[3]{3x}} dx$	$\int_0^1 \frac{dt}{\sqrt{(1+t^2)(4+t^2)}}$		
$\int_0^5 e^x \sin x^2 dx$	$\int_0^1 \sqrt{\ln \frac{1}{t}} dt$	$\int_0^1 \frac{3x^4 + 3x^2 + 1}{x^2 + 1} dx$	
$\int_0^{\sqrt{3}} x \sqrt[3]{1+x^2} dx$	$\int_0^3 y \ln(y - 1) dy$	$\int_0^{12\sqrt{3}} \frac{12x^5}{\sqrt{x^6 + 1}} dx$	
$\int_0^1 \frac{x^2}{x^2 + 1} dx$	$\int_0^{\pi/2} x \cos x dx$	$\int_2^3 \frac{x+2}{x^2(x-1)} dx$	
$\int_0^{\pi/2} \sin x \cos^2 x dx$	$\int_0^{\pi} x^2 \sin x dx$	$\int_2^3 \frac{1}{x^2(x-1)} dx$	
$\int_0^{\pi/2} \frac{\cos x}{1 + \cos x} dx$	$\int_{-1/2}^{1/2} \arccos 2x dx$	$\int_{-1}^1 \frac{y^5}{y+2} dy$	

$\int_{3/4}^{4/3} \frac{1}{x^2 + 1} dx$	$\int_1^2 (y - 1) \ln y dy$	$\int_2^3 \frac{3x^2 + 2x - 3}{x^3 - x} dx$
$\int_{1/3}^{1/2} \frac{x}{(x-1)^3} dx$	$\int_0^2 \frac{x^3}{\sqrt{x^4 + 4}} dx$	$\int_{-\pi}^{\pi} x \sin x \cos x dx$
$\int_4^5 \frac{dx}{(x-1)(x+2)}$	$\int_1^e \frac{1 + \ln x}{x} dx$	$\int_{1/3}^{2/3} \frac{x}{e^{3x}} dx$
$\int_3^4 \frac{dx}{(x+1)(x-2)}$	$\int_1^e \frac{2x+3}{(x-2)^3} dx$	$\int_1^{e^2} \frac{1}{(x-1)^2(x+1)} dx$

2-Topshiriq. Differensial tenglamalarni turli usul va operatorlardan foydalanib yeching

1. Birinchi tartibdagi differensial tenglamalarni berilgan nuqtalardagi qiymatlarni toping va funksiya o'zgarish grafigini chizing

1. $y' = x + \cos \frac{y}{\pi}$, $y(1.7) = 5.3$, $y(2.6) = ?$, $y(5.2) = ?$
2. $y' = \frac{x}{2} + \frac{e^2}{x+y}$, $y(1.8) = 4.5$, $y(2.0) = ?$, $y(3.1) = ?$, $y(4.6) = ?$
3. $y' = \sqrt[3]{x^2 + 3y}$, $y(3) = 5$, $y(4.0) = ?$, $y(8.2) = ?$, $y(11.4) = ?$
4. $y' = \frac{y}{x}(y \ln x - 1)$, $y(1) = 0.5$, $y(1.2) = ?$, $y(0.6) = ?$
5. $y' = 2x + \sin \frac{y}{x}$, $y(0) = 1$, $y(0.3) = ?$, $y(0.5) = ?$, $y(1.0) = ?$
6. $y' = \sqrt{x} + \sqrt{y}$, $y(0) = 0$, $y(0.25) = ?$, $y(0.5) = ?$, $y(1.0) = ?$
7. $y' = x + \sqrt[3]{3 + y^2}$, $y(0) = 0$, $y(0.5) = ?$, $y(1.0) = ?$
8. $y' = \frac{y}{e^{-1} + y^2}$, $y(0) = 1$, $y(0.3) = ?$, $y(0.8) = ?$, $y(1.0) = ?$
9. $y' = x - y$, $y(0) = 0$, $y(0.4) = ?$, $y(0.7) = ?$, $y(1.0) = ?$
10. $y' = \frac{xy}{1-x^2}$, $y(0) = 1$, $y(0.25) = ?$, $y(0.4) = ?$, $y(0.5) = ?$
11. $y' = 2xy$, $y(0) = 1$, $y(0.20) = ?$, $y(0.75) = ?$, $y(1.0) = ?$
12. $y' = 2x - 3y$, $y(0) = 0$, $y(0.15) = ?$, $y(0.75) = ?$, $y(1.0) = ?$
13. $y' = \sqrt{1 + x^3 + y}$, $y(0.8) = 3.80$, $y(1.25) = ?$, $y(5.0) = ?$

14. $y' = x + y^3, y(0) = 0$
15. $y' = 2y^2 - x, y(1) = 1$
16. $y' + 3y = 1, y(0) = 0$
17. $y' - \tan(x) \cdot y = 0, y(0) = 0$
18. $y' - 2y = e^{2x}, y(0) = 1$
19. $y' - 2xy = e^{x^2}, y(1) = 1$
20. $y' - 2 \tan(x) \cdot y = \cos^4(x), y(0) = -1$
21. $(2x + ye^{xy})dx + (1 + xe^{xy})dy = 0, y(0) = 1$
22. $(x^2 + 2xy)dx + (x^2 - y^2)dy = 0, y(1) = -1$
23. $(3x^2 + 10xy)dx + (5x^2 - 1)dy = 0, y(1) = 0$
24. $e^{-y}dx + (2 - xe^{-y})dy = 0, y(1) = 1$
25. $\left(\frac{x}{\sqrt{x^2+y^2}} + y\right)dx + \left(x + \frac{y}{\sqrt{x^2+y^2}}\right)dy = 0, y(\sqrt{2}) = \sqrt{2}$
26. $\frac{dx}{x} + \frac{dy}{y \ln y} = 0, y(e) = e$

3-Topshiriq. Ikkinchi tartibdagi differensial tenglamalarni ixtiyoriy nuqtalardagi qiymatlarni toping va funksiya o'zgarish grafigini chizing

1. $y'' = (e^{2x} + \sin 3x)x, y(0) = 1, y'(0) = 1$
2. $y'' = (x^2 + 7x + 9)e^x, y(0) = 1, y'(0) = 4$
3. $y'' \tan y = 2(y')^2, y(1) = \frac{\pi}{4}, y'(1) = -2$
4. $y'' - 2xy' = 0, y(0) = 1, y'(0) = 1$
5. $y'' = y' \ln y', y(0) = 1, y'(0) = 1$
6. $3y'y'' = 2y, y(0) = 1, y'(0) = 1$
7. $y'' = 2y^3, y(0) = 1, y'(0) = 1$
8. $y'' + y'\sqrt{(y')^2 - 1} = 0, y(\pi) = 0, y'(\pi) = -1$
9. $y'' + 3y' + 2y = 0, y(0) = -1, y'(0) = 3$
10. $y'' + 2y' + 5y = 0, y(0) = 1, y'(0) = 1$
11. $y'' + 4y = 0, y\left(\frac{\pi}{4}\right) = 1, y'\left(\frac{\pi}{4}\right) = -2$
12. $y'' + 4y' + 4y = 0, y(0) = 1, y'(0) = -1$
13. $y'' - 10y' + 25y = 0, y(0) = 2, y'(0) = 8$
14. $y'' + 2y' - 8y = 0, y(0) = 4, y'(0) = -4$
15. $y'' - 2y' + y = 0, y(0) = 4, y'(0) = 2$
16. $y'' - 4y' + 5y = 0, y(0) = 0, y'(0) = 1$
17. $y'' + y = 4xe^x, y(0) = -2, y'(0) = 0$
18. $y'' + y = \sin x, y(0) = 1, y'(0) = 2$

19. $y'' + 4y' + 4y = 8xe^{2x}, y(0) = -1, y'(0) = 1$
 20. $y'' + 9y = 6 \cos 3x, y(0) = 1, y'(0) = 3$
 21. $y'' - y = e^x, y(0) = 0, y'(0) = -1$
 22. $y'' + 2y' - 3y = x^2 e^x, y(0) = 1, y'(0) = 1$

4-Topshiriq. Turli fizik jarayonlarni modellashtirib, jism harakatini yoki uning ko'rsatkichlari o'zgarishini o'zaro yoki vaqtga nisbatan ko'rsating.

1. **Garmonik ossillyator.** Umumiy holda, massaning muvozanat holatiga nisbatan $Z = z - z_0$ joylashishini tavsiflovchi harakat tenglamasi keng ma'lum ikkinchi darajali differensial tenglamadir

$$\frac{d^2 Z(t)}{dt^2} + \frac{b}{m} \frac{dZ(t)}{dt} + \frac{K}{m} Z(t) = \frac{F_{tashqi}(t)}{m} \quad (1)$$

bu yerda b – ishqalanish koeffitsiyenti, $F_{tashqi}(t)$ - sistemaga ta'sir etayotgan tashqi kuch. $F_{tashqi}(t) = 0$ bo'lsa, harakat tenglamasi $Z(t)$ faqat sistemaning m , b va K parametrlariga bog'liq bo'ladi:

$$\frac{d^2 Z}{dt^2} + 2\alpha \frac{dZ}{dt} + \omega_0^2 Z = 0; \quad \alpha = \frac{b}{2m}; \quad \omega_0 = \sqrt{k/m} \quad (2)$$

2. **Rentgen nurlarining refraksiyasi**

$$I(q) = I_0 + \sum_{k=0}^N a_k e^{-(q-q_k)^2/b_k^2} \quad (2)$$

I_0 – boshlang'ich nurlanish; q_k , a_k , b_k – k indeksi bilan belgilangan elementning to'liq vektori, amplitudasi va chiziqlarning kengligi.

3. Jismning qovushqoq muhitdagi harakatim Stoks qonuni bilan ifodalasa bo'ladi $F_S = 6\pi \cdot \eta \cdot r \cdot v$ (v – jismning tezligi, r – uning radiusi, η – muhitning qovushqoqligi koeffitsiyenti). Shunda jism qovushqoq muhitda pastga tushadigan bo'lsa, unga tortishish kuchidan tashqari Stoks va Arximed kuchlari ta'sir qiladi

$$mg + F_A + F_S = 0. \quad (5)$$

Shu ko'rinishdagi differensial tenglamani yechish kerak bo'ladi.

Birinchi yaqinlashishda Arximed kuchini e'tiborga olmasa ham bo'ladi.

4. Lorens kuchi bilan elektr va magnit maydonda harakatlanayotgan zarrachani ko'rsatsa bo'ladi.

$$\text{Magnit maydon: } R = \frac{mv}{Bq}; \quad \omega = \frac{v}{R} = \frac{Bq}{m}; \quad T = \frac{2\pi}{\omega} = 2\pi \frac{m}{Bq}$$

Elektr maydon:

$$a = \frac{F}{m} = \frac{qE}{m}$$

Harakat tenglamalari:

$$x = R \sin \omega t + \frac{at^2}{2}; \quad y = R \cos \omega t + \frac{at^2}{2} \quad (6)$$

$$m_0 = 1.67 \cdot 10^{-27} \text{ kg}; \quad e_0 = 1.6 \cdot 10^{-19} \text{ C}; \quad q = N \cdot e_0; \quad m = N \cdot m_0; \\ E \in [0, 10]; \quad B \in [0, 10]$$

Shu kabi tenglamalar soni va turlari juda ko'p. Mexanika, Molekulyar fizika, Elektr va magnetizm, Optika, Matematik fizika kabi fanlar doirasida uchraydigan fizik hodisa va obyektlarga tegishli tenglamalarni topib, ularni ko'rgazmali ravishda yechsa bo'ladi.

GLOSSARIY (LUG‘AT)

1) **Model** – Haqiqiy dunyodagi jarayonlar, tizimlar yoki hodisalarni soddalashtirilgan shaklda ifodalash vositasi. U matematik, grafik yoki kompyuter dasturlari orqali tuzilishi mumkin.

2) **Modellashtirish** – Murakkab tizimlar yoki fizik jarayonlarni tushunish, tahlil qilish va bashorat qilish uchun ularni soddalashtirilgan shaklda (modelda) tasvirlash jarayoni.

3) **Fizik jarayon** – Tabiatda fizik qonuniyatlar asosida sodir bo‘ladigan hodisalar (masalan, harakat, issiqlik uzatish, elektr maydonlari).

4) **Eksperiment** – Ilmiy tadqiqot uchun maxsus sharoitlarda o‘tkaziladigan sinov yoki tajriba.

5) **Kuzatuv** – Tabiiy yoki sun‘iy hodisalarni bevosita yoki asboblardan yordamida o‘rganish va ma’lumot yig‘ish jarayoni.

6) **Analitik model** – Matematik tenglamalarni aniq yechish orqali tuziladigan model turi. Soddaroq tizimlar uchun qo‘llaniladi.

7) **Sonli model** – Murakkab tizimlarni raqamli usullar (masalan, differensial tenglamalarni hisoblash) yordamida ifodalash.

8) **Modellashtirish bosqichlari** – Masalani tahlil qilish, matematik ifodalash, algoritm ishlab chiqish, dastur yaratish va natijalarni talqin qilish kabi ketma-ket jarayonlar.

9) **Matematik operatorlar** – Sonlar va o‘zgaruvchilar ustida amallar bajarish uchun ishlatiladigan belgilar (masalan, +, -, *, /).

10) **Mantiqiy operatorlar** – Shartlarni tekshirish uchun ishlatiladigan operatorlar (masalan, ==, >, <).

11) **Resurslar** – Modellashtirish jarayonida ishlatiladigan vaqt, mablag‘, ma’lumotlar va texnik vositalar.

12) **Python dasturlash tili** – Oddiy sintaksisga ega, ko‘p maqsadli, yuqori darajali dasturlash tili. Modellashtirish va ilmiy hisoblash uchun keng qo‘llaniladi.

13) **Matlab sistemasi** – Matritsalar asoslangan ilmiy va muhandislik masalalarni yechish uchun mo‘ljallangan dasturiy muhit.

14) **Modellar turlari** – Analitik, sonli, kompyuter, grafik kabi modellashtirishning turli shakllari.

15) **Algoritm** – Ma’lum bir masalani yechish uchun aniq va ketma-ket bajariladigan harakatlar ketma-ketligi.

16) **Blok-sxema** – Algoritmni grafik shaklda tasvirlash usuli. Maxsus belgilar yordamida jarayonlarni ko‘rsatadi.

17)**Chiziqli algoritm** – Buyruqlar ketma-ket bir marta bajariladigan eng sodda algoritm turi.

18)**Sintaksis** – Dasturlash tilida kod yozish qoidalari va tuzilishi.

19)**Kuzatuv ma'lumotlari** – Tabiiy yoki eksperimental jarayonlardan yig'ilgan ma'lumotlar.

20)**Bashorat qilish** – Modellar yordamida kelajakdagi hodisalarni oldindan taxmin qilish.

21)**Simulyatsiya** – Fizik jarayonlarni kompyuter dasturlari orqali taqlid qilish.

22)**Monte-Karlo usuli** – Tasodifiy sonlardan foydalangan holda murakkab tizimlarni modellashtirish usuli.

23)**Molekulyar dinamika** – Molekulalar harakatini modellashtirish uchun ishlatiladigan usul.

24)**Kvant mexanikasi** – Atom va subatom darajasidagi jarayonlarni o'rganadigan fizika sohasi.

25)**Nisbiylik nazariyasi** – Vaqt, fazo va gravitatsiyani tavsiflovchi Albert Eynshteynning nazariyasi.

26)**Gidrodinamika** – Suyuqlik va gazlar harakatini o'rganadigan fizika bo'limi.

27)**Radiatsiya uzatish** – Nur yoki elektromagnit to'lqinlarning tarqalishini o'rganadigan soha.

28)**Interpretator** – Python kabi tillarda kodni kompilyatsiya qilmasdan to'g'ridan-to'g'ri ishga tushiradigan dastur.

29)**Dinamik tipizatsiya** – O'zgaruvchilar turini oldindan aniqlamasdan ishlatish imkoniyati.

30)**Kutubxona** – Dasturlash tilida muayyan vazifalarni bajarish uchun tayyor funksiyalar to'plami (masalan, NumPy, Pandas).

31)**Indentatsiya** – Python'da kod bloklarini ajratish uchun ishlatiladigan joy tashlash qoidasi.

32)**MATLAB interfeysi** – Foydalanuvchi va MATLAB tizimi o'rtasidagi muloqot vositasi (buyruqlar oynasi, ishchi soha va boshqalar).

33)**Spyder muhiti** – Python dasturlash uchun ilmiy hisoblashga mo'ljallangan maxsus dasturiy muhit.

34)**Debug vositasi** – Kod xatolarini aniqlash va tuzatish uchun ishlatiladigan vosita.

35)**Arifmetik operatorlar** – Matematik amallarni bajarish uchun ishlatiladigan belgilar (qo'shish, ayirish, ko'paytirish va hokazo).

36)**Belgilash operatorlari** – O'zgaruvchilarga qiymat berish yoki ularni o'zgartirish uchun ishlatiladigan operatorlar (masalan, =, +=).

37)**Trigonometrik funksiyalar** – Sinus, kosinus, tangens kabi burchaklarni hisoblash uchun ishlatiladigan matematik funksiyalar.

38)**Tasodifiy sonlar** – Random moduli yordamida tasodifiy qiymatlar olish.

39)**Ma'lumot turlari** – Dasturlashda ishlatiladigan qiymat turlari (string, integer, float, boolean va boshqalar).

40)**Funksiya** – Kodni qayta ishlatish uchun maxsus nomlangan va ma'lum vazifani bajaradigan kod bloki.

41)**Tarmoqlanuvchi algoritm** – Bir yoki bir nechta shartlarga qarab turli yo'nalishlarda (tarmoqlarda) bajariladigan buyruqlar ketma-ketligi bo'lgan algoritm turi.

42)**Mantiqiy operatsiyalar** – Bayonotlarni (haqiqat yoki yolg'on qiymatlarni) birlashtirish yoki o'zgartirish uchun ishlatiladigan amallar (masalan, AND, OR, NOT).

43)**Sikllar** – Dasturda buyruq yoki buyruqlar guruhini bir necha marta takrorlash uchun ishlatiladigan tuzilmalar.

44)**Siklik operatorlar** – Sikllarni amalga oshirish uchun ishlatiladigan maxsus operatorlar (masalan, for va while).

45)**Munosabatlar operatorlari** – Ikkita kattalikni solishtirish uchun ishlatiladigan belgilar (masalan, ==, !=, >, <, >=, <=).

46)**Tenglik (==)** – Ikkita kattalikning bir-biriga tengligini tekshiradigan operator.

47)**Teng emaslik (!=)** – Ikkita kattalikning bir-biriga teng emasligini tekshiradigan operator.

48)**Katta (>)** – Bir kattalikning ikkinchisidan katta ekanligini tekshiradigan operator.

49)**Kichik (<)** – Bir kattalikning ikkinchisidan kichik ekanligini tekshiradigan operator.

50)**Katta yoki teng (>=)** – Bir kattalikning ikkinchisidan katta yoki teng ekanligini tekshiradigan operator.

51)**Kichik yoki teng (<=)** – Bir kattalikning ikkinchisidan kichik yoki teng ekanligini tekshiradigan operator.

52)**Shartli operatorlar** – Shartga qarab buyruqlarni bajarishni boshqaradigan operatorlar (masalan, if, else, elseif, switch).

53)**Chala tarmoqli algoritm** – Faqat bitta tarmoqqa ega bo'lgan tarmoqlanuvchi algoritm.

54)**To'liq tarmoqli algoritm** – Ikkita tarmoqqa (shart bajarilsa va bajarilmasa) ega bo'lgan algoritm.

55)**Ko'p tarmoqli algoritm** – Ikkitadan ko'p tarmoqqa ega bo'lgan algoritm.

56)**Konyuksiya (AND, &)** – Mantiqiy ko'paytirish; ikkala bayonot ham to'g'ri bo'lsa, natija to'g'ri bo'ladi.

57)**Dizyunksiya (OR, |)** – Mantiqiy qo'shish; kamida bitta bayonot to'g'ri bo'lsa, natija to'g'ri bo'ladi.

58)**Inversiya (NOT, ~)** – Mantiqiy rad etish; bayonotning haqiqat qiymatini teskarisiga o'zgartiradi.

59)**Bayonot** – “To'g'ri” yoki “yolg'on” deb baholanishi mumkin bo'lgan ifoda yoki gap.

60)**Operand** – Mantiqiy operatsiyada ishlatiladigan bayonot yoki qiymat (1 yoki 0).

61)**Haqiqat (True)** – Mantiqiy ifodaning to'g'ri ekanligini bildiruvchi qiymat (1).

62)**Yolg'on (False)** – Mantiqiy ifodaning noto'g'ri ekanligini bildiruvchi qiymat (0).

63)**For sikli** – Oldindan belgilangan marta takrorlanadigan siklik operator.

64)**While sikli** – Shart to'g'ri bo'lguncha takrorlanadigan siklik operator.

65)**Indeks** – Siklda takrorlanish sonini boshqaradigan o'zgaruvchi (sanagich).

66)**Diapazon** – Siklda indeksning qabul qiladigan qiymatlari oralig'i.

67)**Break operatori** – Siklni shart bajarilganda to'xtatadigan operator.

68)**Continue operatori** – Siklda joriy qadamni o'tkazib yuborib, keyingi qadamga o'tadigan operator.

69)**Pause operatori** – Siklni vaqtincha to'xtatadigan operator.

70)**Ichki sikllar** – Bir sikl ichida joylashgan boshqa sikl(lar).

71)**Faktorial** – Berilgan sonning barcha undan kichik musbat sonlarga ko'paytmasi (masalan, $5! = 5 \times 4 \times 3 \times 2 \times 1$).

72)**Eng katta umumiy bo'luvchi (EKUB)** – Ikki sonning umumiy bo'luvchilari ichida eng kattasi.

73)**Evklid usuli** – Ikki sonning eng katta umumiy bo'luvchisini topish uchun ishlatiladigan algoritm.

74)**Taylor qatori** – Funksiyani ma'lum nuqtada polinomlar yig'indisi sifatida ifodalash usuli.

75)**Aniqlik (ϵ)** – Hisoblashda qo'llaniladigan xatolik chegarasi.

76)**Blok-sxema** – Algoritmni grafik shaklda tasvirlash usuli, maxsus belgilar yordamida jarayonlarni ko'rsatadi.

- 77)**Sintaksis** – Dasturlash tilida kod yozish qoidalari va tuzilishi.
- 78)**Massiv** – Bir xil turdagi ma'lumotlarni (sonlar, matnlar, obyektlar) ketma-ket saqlash uchun ishlatiladigan ma'lumotlar strukturasi. Har bir element indeks orqali chaqiriladi.
- 79)**Indeks** – Massivdagi elementlarga murojaat qilish uchun ishlatiladigan raqamli ko'rsatkich, odatda 0 dan boshlanadi.
- 80)**Ro'yxat (List)** – Pythonda massivga o'xshash dinamik ma'lumotlar strukturasi, turli turdagi elementlarni saqlashi mumkin.
- 81)**NumPy massivi** – NumPy kutubxonasi orqali yaratiladigan, yuqori samaradorlikdagi aniq massiv turi, odatda sonli ma'lumotlar bilan ishlaydi.
- 82)**Bir o'lchamli massiv** – Elementlari bitta chiziq (ro'yxat) shaklida joylashgan massiv (masalan, [1, 2, 3]).
- 83)**Ko'p o'lchamli massiv** – Matritsa yoki ko'p o'lchamli shaklda joylashgan massiv (masalan, 2D: [[1, 2], [3, 4]]).
- 84)**Xotira** – Massiv elementlari ketma-ket joylashadigan kompyuterning saqlash maydoni.
- 85)**Tez kirish (O(1))** – Massiv elementlariga indeks orqali doimiy vaqt ichida murojaat qilish imkoniyati.
- 86)**Append()** – Ro'yxat oxiriga yangi element qo'shish uchun ishlatiladigan metod.
- 87)**Insert()** – Ro'yxatning ixtiyoriy indeksiga yangi element qo'shish uchun ishlatiladigan metod.
- 88)**Remove()** – Ro'yxatdan ma'lum bir qiymatni o'chirish uchun ishlatiladigan metod.
- 89)**Pop()** – Ro'yxatdan indeks bo'yicha elementni o'chirib, uni qaytaradigan metod.
- 90)**Slicing (Kesib olish)** – Ro'yxatning bir qismini indekslar oralig'i orqali ajratib olish usuli (masalan, [1:3]).
- 91)**Len()** – Ro'yxat yoki massiv uzunligini (elementlar sonini) qaytaradigan funksiya.
- 92)**Sort()** – Ro'yxatni tartiblash uchun ishlatiladigan metod (o'sish yoki kamayish tartibida).
- 93)**Grafik** – Ma'lumotlarni vizual shaklda (chiziqlar, nuqtalar, gistogrammalar) ifodalash usuli.
- 94)**Matplotlib** – Pythonda grafiklar chizish uchun ishlatiladigan kutubxona (matplotlib.pyplot moduli).
- 95)**Plot()** – Matplotlibda chiziqli grafik chizish uchun ishlatiladigan funksiya.

96) **Hold on** – Bir nechta grafikni bitta rasmda ko‘rsatish rejimini yoqadigan operator.

97) **Hold off** – Bir nechta grafikni bitta rasmda ko‘rsatish rejimini o‘chiradigan operator.

98) **Grid on** – Grafikda to‘r (setka) ko‘rsatishni yoqadigan operator.

99) **Grid off** – Grafikdagi to‘rni o‘chiradigan operator.

100) **Chiziq ko‘rinishi (Line Style)** – Grafik chiziqlarning shakli (masalan, uzluksiz, shtrixli, punktirli).

101) **Nuqta belgisi (Marker)** – Grafikdagi ma‘lumot nuqtalarini belgilash uchun ishlatiladigan shakl (masalan, aylana, yulduzcha).

102) **Rang (Color)** – Grafik chiziq yoki nuqtalarining rangi (masalan, qizil, ko‘k, sariq).

103) **RGB uchtaligi** – Rangni qizil (Red), yashil (Green), ko‘k (Blue) komponentlari orqali $[0, 1]$ oralig‘ida belgilash usuli.

104) **Linewidth** – Grafik chiziqning qalinligini belgilovchi parametr.

105) **Markersize** – Grafik nuqtalarining o‘lchamini belgilovchi parametr.

106) **MarkerEdgeColor** – Nuqtaning chetki chizig‘i rangini belgilovchi parametr.

107) **MarkerFaceColor** – Nuqtaning ichki qismining rangini belgilovchi parametr.

108) **Title()** – Grafikning umumiy nomini (sarlavhasini) qo‘yish uchun ishlatiladigan funksiya.

109) **Xlabel()** – Gorizontal o‘qqa izoh (yorliq) qo‘yish uchun ishlatiladigan funksiya.

110) **Ylabel()** – Vertikal o‘qqa izoh (yorliq) qo‘yish uchun ishlatiladigan funksiya.

111) **Legend()** – Grafikdagi chiziqlarning izohini ko‘rsatadigan funksiya.

112) **Axis()** – Grafik o‘qlarining chegaralarini belgilash uchun ishlatiladigan funksiya (masalan, $[-5, 5, -1, 1]$).

113) **Text()** – Grafikda ixtiyoriy joyga matnli izoh qo‘shish uchun ishlatiladigan funksiya.

114) **Subplot()** – Bitta rasmda bir nechta grafikni joylashtirish uchun ishlatiladigan funksiya (satr, ustun, tartib raqami).

115) **Savefig()** – Grafikni fayl (masalan, PNG, JPG, PDF) sifatida saqlash uchun ishlatiladigan funksiya.

- 116) **3D grafik** – Uch o‘lchamli fazoda (x, y, z) chizilgan grafik turi.
- 117) **Gistogramma** – Ma’lumotlarning taqsimotini ko‘rsatadigan ustunli grafik turi.
- 118) **Bins** – Gistogrammadagi ustunlar sonini belgilovchi parametr.
- 119) **Edgecolor** – Gistogramma ustunlarining chetki chiziq rangini belgilovchi parametr.
- 120) **Tight_layout()** – Grafik elementlarini avtomatik tartibga soladigan funksiya.
- 121) **Integral** – Funksiyaning ma’lum bir oraliqdagi yig‘indisini hisoblash jarayoni; differensiallashning teskari amali.
- 122) **Integrallash** – Funksiya ostidagi maydon, hajm yoki fizik miqdorlarni (masalan, ish, energiya) hisoblash uchun kichik qismlarni yig‘ish jarayoni.
- 123) **Integrallash chegaralari** – Integrallash jarayonining boshlanish (a) va tugash (b) nuqtalari.
- 124) **Differensial** – O‘zgaruvchining kichik o‘zgarishi (masalan, dx yoki dt).
- 125) **Maydon** – Funksiya grafigi ostidagi chegaralangan hudud; integrallash orqali hisoblanadi.
- 126) **Ish** – Kuchning masofaga bog‘liq ta’sirini hisoblash uchun ishlatiladigan fizik miqdor ($W = \int F(x) dx$).
- 127) **Energiya** – Jism yoki tizimning holatini tavsiflovchi miqdor, ko‘pincha integrallash orqali aniqlanadi.
- 128) **Masofa** – Tezlikni vaqt bo‘yicha integrallash orqali hisoblanadigan fizik miqdor ($s = \int v(t) dt$).
- 129) **Zaryad** – Elektr tokini vaqt bo‘yicha integrallash orqali topiladigan miqdor ($q = \int I(t) dt$).
- 130) **To‘g‘ri burchaklar usuli** – Integrallashda diapazonni kichik bo‘laklarga bo‘lib, har bir bo‘lakning balandligini yig‘ish usuli.
- 131) **Trapetsiya usuli** – Integrallashda diapazonni trapetsiyalarga bo‘lib, birinchi va oxirgi hadlarning yarmini hisobga oladigan usul.
- 132) **Simpson usuli** – Integrallashda diapazonni parabolalar yordamida taqriban hisoblash usuli; toq indeksli hadlar 4 ga, juft indeksli hadlar 2 ga ko‘paytiriladi.
- 133) **Integrallash qadami (h)** – Integrallash diapazonini bo‘lishda ishlatiladigan kichik interval ($h = (b - a) / n$).
- 134) **Differensial tenglama** – Funksiya, uning argumentlari va hosilalarini o‘z ichiga olgan tenglama.

- 135) **Oddiy differensial tenglama** – Bitta mustaqil o‘zgaruvchiga bog‘liq bo‘lgan differensial tenglama.
- 136) **Xususi differensial tenglama** – Bir nechta mustaqil o‘zgaruvchilarga bog‘liq bo‘lgan differensial tenglama.
- 137) **Tartib** – Differensial tenglamada ishtirok etadigan eng yuqori hosilaning darajasi.
- 138) **Boshlang‘ich shart** – Differensial tenglamaning yagona yechimini aniqlash uchun qo‘yiladigan shart (masalan, $y(t_0) = y_0$).
- 139) **Koshi masalasi** – Differensial tenglamaning boshlang‘ich shart bilan berilgan yechimini topish masalasi.
- 140) **Eyler usuli** – Birinchi tartibdagi differensial tenglamani sonli yechish usuli; har qadamda yechimni qadam bo‘yicha yangilaydi.
- 141) **Runge-Kutta usuli** – Differensial tenglamalarni yuqori aniqlik bilan sonli yechish usuli; bir nechta oraliq qiymatlarni hisoblaydi.
- 142) **Harakat qonuni** – Jismning vaqt bo‘yicha joylashuvi ($x(t)$) yoki koordinatalarini ($x(t), y(t), z(t)$) aniqlovchi tenglama.
- 143) **Mexanik harakat** – Jismning fazoda tortishish yoki boshqa kuchlar ta’sirida harakatlanishi.
- 144) **Erkin tushish tezlanishi (g)** – Yer sathiga yaqin joylashgan jismlarga ta’sir qiluvchi gravitatsion tezlanish (taxminan 9.8 m/s^2).
- 145) **Boshlang‘ich tezlik (v_0)** – Harakat boshlanishidagi jismning tezligi.
- 146) **Burchak (α)** – Jismning gorizontga nisbatan otish burchagi.
- 147) **Potensial energiya** – Jismning balandligiga bog‘liq bo‘lgan energiya ($P = mgy$).
- 148) **Kinetik energiya** – Jismning harakatiga bog‘liq energiya ($K = \frac{1}{2}mv^2$).
- 149) **To‘liq energiya** – Potensial va kinetik energiyalarning yig‘indisi ($E = P + K$).
- 150) **Simvolli amallar** – Matematik ifodalarni analitik (algebraik) ko‘rinishda yechish jarayoni.
- 151) **SciPy** – Pythonda integrallash va differensial tenglamalarni yechish uchun ishlatiladigan kutubxona.
- 152) **SymPy** – Pythonda simvolli matematik amallarni bajarish uchun ishlatiladigan kutubxona.
- 153) **ode45** – MATLAB’da Runge-Kutta usuliga asoslangan differensial tenglamalarni yechish operatori.

154) **Tugunlar** – Integrallash yoki differensial tenglamalarni yechishda diapazonni bo‘lish nuqtalari.

155) **Aniqlik** – Sonli usullarda natijaning haqiqiy qiymatga yaqinligi; qadam uzunligiga bog‘liq.

ADABIYOTLAR RO'YXATI

I.O'zbekiston Respublikasi Prezidenti asarlari

1. Mirziyoyev Sh.M. Milliy taraqqiyot yo'limizni qat'iyat bilan davom ettirib, yangi bosqichga ko'taramiz. – T.: “O'zbekiston”, 2017. – 592 b.
2. Mirziyoyev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. – T.: “O'zbekiston”, 2017. – 488 b.
3. Mirziyoyev Sh.M. Milliy taraqqiyot yo'limizni qat'iyat bilan davom ettirib, yangi bosqichga ko'taramiz. 1-jild. – T.: “O'zbekiston”, 2018. – 592 b.
4. Mirziyoyev Sh.M. Xalqimizning roziligi bizning faoliyatimizga berilgan eng oliy bahodir. 2-jild. – T.: “O'zbekiston”, 2019. – 400 b.
5. Mirziyoyev Sh.M. Niyati ulug' xalqning ishi ham ulug', hayoti yorug' va kelajagi farovon bo'ladi. 3-jild. – T.: “O'zbekiston”, 2019. – 592 b.
6. Mirziyoyev Sh.M. Milliy tiklanishdan – milliy yuksalish sari. 4-jild. – T.: “O'zbekiston”, 2020. – 400 b.

II. Normativ-huquqiy hujjatlar

1. O'zbekiston Respublikasining Konstitutsiyasi. – T.: “O'zbekiston”, 2023.
2. O'zbekiston Respublikasining “Ta'lim to'g'risida”gi Qonuni. – 2020-yil 23-sentabr.
3. O'zbekiston Respublikasining “Korrupsiyaga qarshi kurashish to'g'risida”gi Qonuni.
4. O'zbekiston Respublikasi Prezidentining PF-4732-sonli Farmoni. “Oliy ta'lim muassasalarining rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini yanada takomillashtirish to'g'risida” – T.: O'zbekiston, 2015-yil 12-iyun.
5. O'zbekiston Respublikasi Prezidentining PF-5729-sonli Farmoni. “O'zbekiston Respublikasida korrupsiyaga qarshi kurashish tizimini yanada takomillashtirish chora-tadbirlari to'g'risida” – T.: O'zbekiston, 2019-yil 27-may.
6. O'zbekiston Respublikasi Prezidentining PF-5789-sonli Farmoni. “Oliy ta'lim muassasalari rahbar va pedagog kadrlarining uzluksiz malakasini oshirish

- tizimini joriy etish to'g'risida" – T.: O'zbekiston, 2019-yil 27-avgust.
7. O'zbekiston Respublikasi Vazirlar Mahkamasining 797-sonli Qarori. "Oliy ta'lim muassasalari rahbar va pedagog kadrlarining malakasini oshirish tizimini yanada takomillashtirish bo'yicha qo'shimcha chora-tadbirlar to'g'risida" – T.: O'zbekiston, 2019-yil 23-sentabr.
 8. O'zbekiston Respublikasi Prezidentining PF-5847-sonli Farmoni. "O'zbekiston Respublikasi oliy ta'lim tizimini 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to'g'risida" – T.: O'zbekiston, 2019-yil 8-oktabr.
 9. O'zbekiston Respublikasi Prezidentining PF-6097-sonli Farmoni. "Ilm-fanni 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to'g'risida" – T.: O'zbekiston, 2020-yil 29-oktabr.
 10. O'zbekiston Respublikasi Prezidentining PQ-4996-son Qarori. "Sun'iy intellekt texnologiyalarini jadal joriy etish uchun shart-sharoitlar yaratish chora-tadbirlari to'g'risida" – T.: O'zbekiston, 2021-yil 17-fevral.
 11. O'zbekiston Respublikasi Prezidentining PF-60-son Farmoni. "2022–2026 yillarga mo'ljallangan Yangi O'zbekistonning taraqqiyot strategiyasi to'g'risida" – T.: O'zbekiston, 2022-yil 28-yanvar.
 12. O'zbekiston Respublikasi Prezidentining PF-14-sonli Farmoni. "Respublika ijro etuvchi hokimiyat organlari faoliyatini samarali yo'lga qo'yishga doir birinchi navbatdagi tashkiliy chora-tadbirlar to'g'risida" – T.: O'zbekiston, 2023-yil 25-yanvar.
 13. O'zbekiston Respublikasi Prezidentining PF-158-son Farmoni. "O'zbekiston – 2030" strategiyasi to'g'risida" – T.: O'zbekiston, 2023-yil 11-sentabr.
 14. O'zbekiston Respublikasi Prezidentining PQ-228-son Qarori. "Aholi va davlat xizmatchilarining korrupsiyaga qarshi kurashish sohasidagi bilimlarini uzluksiz oshirish tizimini joriy qilish chora-tadbirlari to'g'risida" – T.: O'zbekiston, 2024-yil 21-iyun.

III. Maxsus adabiyotlar

1. **Schmidt W., Völschow M.** *Numerical Python in Astronomy and Astrophysics: A Practical Guide to Astrophysical Problem Solving.* – Springer, 2021. – 257 p. – ISBN 978-3-030-70346-2.
2. **Axmadjonov T., Akramov M.** *Fizik jarayonlarni kompyuterda modellashtirish.* O‘quv qo‘llanma. – Toshkent: “Universitet” nashriyoti, 2024.
3. **Akramov M., Axmadjanov T.** *Fizik jarayonlarni kompyuterda modellashtirish.* O‘quv qo‘llanma. – Toshkent: “Universitet” nashriyoti, 2024.
4. **Нагаева И.А., Кузнецов И.А.** *Основы математического моделирования и численные методы.* Учебное пособие. – Санкт-Петербург: Лань, 2024. – 204 с.
5. **Bodenheimer P., Laughlin G., Rozyczka M., Yorke H.W.** *Numerical Methods in Astrophysics: An Introduction (Series in Astronomy and Astrophysics).* – Boca Raton: CRC Press, 2007. – 330 p. – ISBN: 978-0-7503-0883-0.
6. **Слабнов В. Д.** *Численные методы.* Учебное пособие. – Москва: Лань, 2024. – 392 с.

IV. Internet saytlar

1. O‘zbekiston Respublikasi Oliy ta’lim, fan va innovatsiyalar vazirligi rasmiy veb-sayti – www.edu.uz
2. O‘zbekiston Respublikasi Akkreditatsiya markazi rasmiy sayti – www.aci.uz
3. Axborot-kommunikatsiya texnologiyalarini rivojlantirish kengashi rasmiy sayti – www.ictcouncil.gov.uz
4. BIMM Elektron kutubxonasi – www.lib.bimm.uz
5. Ta’lim va ilm-fan bo‘yicha milliy portal "ZiyoNET" – www.ziynet.uz
6. ScienceDirect – Ilmiy maqolalar va tadqiqot bazasi – www.sciencedirect.com