

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ
АЛОҚА ВА АХБОРОТЛАШТИРИШ АГЕНТЛИГИ

Тошкент ахборот технологиялари университети
Самарқанд филиали

Урунбаев Э.У., Абдуллаева Н.И., Ишонкулов А.К.

Компьютер алгебраси тизимлари
маърузалар матни

*ТАТУ Самарқанд филиали ўқув –
услубий кенгаши томонидан
ўқув мақсадларида
фойдаланишга тавсия этилган*

САМАРҚАНД - 2011

Урунбаев Э.У., Абдуллаева Н.И. Компьютер алгебраси тизимлари: Маърузалар матни. – Самарқанд: ТАТУ Самарқанд филиали, 2011. – 80 бет.

«Компьютер алгебраси тизимлари» маърузалар матни «Компьютер алгебраси тизимлари» фани дастури асосида тузилган. Маърузалар матнида компьютер алгебраси тизимлари, Maple тизими командасининг синтаксиси, тенглама ва тенгсизликларни ечиш, чизикли алгебра амалларини бажариш, дифференциаллаш ва интеграллаш командалари, Maple тизимнинг график структуралари, функцияларни текшириш, кўп ўзгарувчили функцияларнинг дифференциал ва интеграл ҳисоби, қаторлар устида амаллар, тизимининг дастурлаштириш воситалари кенг ёритилган.

Ушбу маърузалар матни ОТМ 5521900 - Информатика ва ахборот технологиялари бакалавриат йўналишлари учун мўлжалланган.

Такризчилар: СамДУ Математик моделлаштириш
кафедраси доценти ф.-м.ф.н. Ашуров А
ТАТУ СФ Ўқув ва илмий ишлар бўйича
директор ўринбосари ф.-м.ф.н Нарзуллаев У.Х.

Кириш

Ҳозирги пайтда компьютер алгебраси тизимлари математика, физика, химия, информатика каби кўплаб фан соҳаларида, техникада, таълим соҳасида кенг қўлланилмоқда.

Maple, Mathematica, Reduce, MuPAD, Derive, Magma, Macsyma, Axiom каби компьютер алгебраси тизимлари математикага йўналтирилган фанларни ўқитишда, илмий изланишларда ва ишлаб чиқаришда оммавийлашиб бормоқда. Бу тизимлар педагог, мухандис ва олимлар учун қулай восита ҳисобланади. Компьютер алгебраси тизимлари асосида изланишлар олиб бориш алгебраик усулларни илғор ҳисоблаш усуллари билан бирлаштиради. Шу маънода компьютер алгебраси тизимлари математика ва информатика ўртасидаги оралиқ фан бўлиб, унинг мақсади алгебраик ҳисоблашлар алгоритмларини ишлаб чиқиш ва компьютерда қайта ишлаш, бу алгоритмларни қўллаш учун дастурлаш тиллари ва воситаларини яратиш ҳамда уларга асосланган турли хил масалаларни ечишга қаратилади.

Maple пакети тадқиқотчилар учун турли хил масалаларни ечишда муҳим восита ҳисобланади. Maple турли математик моделларни ифодалаш, ечиш ва тадқиқ қилишда тенгсиз ҳисобланади. Унинг алгебраик воситалари юқори савияда ечиш мумкин бўлган муаммолар доирасини кенгайтиради. Мактаб, коллеж, университет ўқитувчилари математик ва физик машқ-масалаларни Maple да ечиш билан одатий ўқитиш услубларини янгиламоқда. Натижада талабалар эътиборларини кўп вақт талаб қиладиган алгебраик ҳисобларга эмас балки масаланинг муҳим жихатига қаратмоқда.

1- Мавзу: Компьютер алгебраси тизимлари тўғрисида умумий тушунчалар. Maple тизими график интерфейсининг таркибий қисмлари

Режа:

1. Компьютер алгебраси тизимлари тўғрисида умумий тушунчалар
2. Maple ойнасининг структураси.
3. Maple да арифметик операциялар, бутун ва рационал сонлар ва константалар.
4. Командалар синтаксиси. Стандарт функциялар.

1. Компьютер алгебраси тизимлари тўғрисида умумий тушунчалар

Профессионал математик пакетлар деганда, одатда, ҳозирги замон компьютерларида қўлланилаётган Mathematica, Maple, Matlab, Derive, Mathcad каби тизимлар ва қийматларни статистик таҳлил қилишга мўлжалланган SPSS, Statistica, Statgraphics, Stadia ва шунга ўхшаш тизимлар оилалари назарда тутилади. Бу пакетлар ёрдамида турли сонли ва аналитик (символли) математик ҳисобларни, оддий арифметик ҳисоблашлардан тортиб, то хусусий ҳосилали дифференциал тенгламаларни ечиш, оптималлаш масалаларини ҳал қилиш, статистик гипотезаларни текшириш ҳамда математик моделларни яратишга қадар турли зарур техник ҳисоблашларни амалга ошириш мумкин. Уларнинг барчаси такомиллашган илмий графика воситаларига, қулай ёрдамчи ахборот тизими ва ҳисоботларни расмийлаштириш воситаларига эга. «Профессионал пакет» атамаси «ўқув пакети» атамасига муқобил равишда танланган.

Ҳозирги даврда бу пакетлардан нафақат олий малакали мутахассислар, профессор-ўқитувчилар, балки олий мактаб талабалари, ҳаттоки, лицейлар, касб-ҳунар коллежлари ва мактаблар ўқувчилари ҳам фойдаланаётганлигини кузатиш мумкин.

2. Maple ойнасининг структураси

Maple – бу компьютерда аналитик ҳисоблашлар бажариш учун пакет, ўзида алгебра, геометрия, математик анализ, дифференциал тенгламалар, статистика, математик физика масалаларини ечиш имконини берувчи икки мингдан ортиқ командаларни сақлайди.

*Maple*ни ишга тушириш учун *Windows* нинг *Бош меню* сидаги *Программы* бандидан *Maple* ни танлаймиз.

Maple Windows нинг типик ойналарига ўхшайди. *Maple* ойнаси Сарлавҳа сатри (*Строки названия*), Бош меню (*Основной меню*), Асбоблар панели (*Панели инструментов*), Ишчи майдон (*Рабочее поле*), Ҳолатлар сатри (*Строка состояния*), Чизғич (*Линейки*), шунингдек, Ўтказиш йўлакчаси (*Полоса прокрутки*) дан ташкил топган.

Maple ойнасининг Сарлавҳа сатри, Бош меню, Асбоблар панелининг кўриниши:



Бош меню таркиби:

File (Файл) – файллар билан ишлаш учун стандарт буйруқлар йиғиндисидан иборат, масалан: файлни сақлаш, файлни очиш, янги файл яратиш ва ҳ.к.

Edit (Правка) – матнни таҳрирлаш учун стандарт буйруқлар йиғиндисидан иборат, масалан: нусха олиш, белгиланган матнни қирқиб олиш, буйруқни рад етиш ва ҳ.к.

View (Вид) – *Maple* ойнаси структурасини бошқарувчи буйруқлар йиғиндисидан иборат.

Insert (Вставка) – турли типдаги майдонларни киритиш, математик палитралар, икки ва уч ўлчамли график тасвирлар учун хизмат қилади.

Format (Формат) – ҳужжатни тузиш учун хизмат қиладиган буйруқлар йиғиндисидан иборат, масалан: шрифтнинг турини, ўлчамини, стилини танлаш.

Options (Параметры) – маълумотларни экранга, принтерга киритиш ва чиқариш параметрларини ростлаш учун мўлжалланган буйруқлар йиғиндисидан иборат, масалан, чоп қилиш сони.

Windows (Окно) – бир ишчи варақдан бошқасига ўтиш учун хизмат қилади.

Help (Справка) – *Maple* ҳақида тўлиқ маълумотларни сақлайди.

Maple да иш сессия тартибида ўтади – фойдаланувчи матн (буйруқлар, ифодалар, процедуралар) ни киритади, улар шартли равишда ҳисобланиб, *Maple* билан қайта ишланади. Ишчи майдон учга бўлинади:

1) Киритиш соҳаси – буйруқлар сатридан иборат. Ҳар бир буйруқ сатри > белгиси билан бошланади;

2) Чиқариш соҳаси – киритилган буйруқларнинг қайта ишланганининг натижаларини аналитик ифодалар, график объектлар ёки хатолар хабарлари кўринишида чиқаради;

3) Матнли изохлар соҳаси – бажарилаётган процедураларни изохлайдиган турли ҳил матнли маълумотларни чиқаради. Матнли сатрлар *Maple* томонидан қабул қилинмайди ва қайта ишланмайди.

Буйруқ сатрини матнли сатрга ўтказиш учун асбоблар панелидаги  тугмани сичқонча билан босиш керак.

Матнли сатрни буйруқ сатрига қайта ўтказиш учун асбоблар панелидаги  тугмани сичқонча билан босиш керак.

3. Арифметик амаллар. *Maple* тизимининг ўзгармаслари, бутун ва рационал сонлар

Математик ўзгармаслар ва арифметик амаллар.

Асосий математик ўзгармаслар: **Pi** – π сони; **I** – мавҳум бирлик i ; **infinity** – чексизлик; **Gamma** – Эйлер ўзгармаси; **true**, **false** – мантикий ифодаларнинг чин ёки ёлғонлигини билдирувчи мантикий ўзгармаслар.

Арифметик амаллар белгилари:

+ - қўшиш; – - айириш;

* - кўпайтириш; / - бўлиш;

^ - даражага кўтариш; ! – факториал.

Такқослаш белгилари: <, >, >=, <=, <>, =

Комплекс, бутун ва рационал сонлар. *Maple* да сонлар ҳақиқий (real) ва комплекс (complex) сонларга бўлинади. Комплекс сонлар $z=x+iy$ алгебраик кўринишда ёзилади, буйруқлар сатрида қуйидагича киритилади:

> z:=x+I*y;

Ҳақиқий сонлар бутун ва рационал сонларга бўлинади. Бутун сонлар (integer) ўнлик системада ёзилади. Рационал сонлар уч хил кўринишда берилиши мумкин:

1) Рационал каср кўринишида, масалан: 28/70;

2) Сузувчи вергул ёрдамида (float), масалан: 2.3;

3) Кўрсаткичли шаклда, масалан: $1,602 \cdot 10^{(-19)}$ сони $1,602 \cdot 10^{-19}$ ни

билдиради.

Рационал соннинг тақрибий қийматини олиш учун сонларнинг бутун қисмига .0 қўшиб ёзиш керак. Мисол:

> 75/4;

$$\frac{75}{4}$$

> 75/4.0;

18.75000000

Maple тизимида грек алфавити ҳарфларини полиграфик кўринишда ёзиш мумкин. Бунинг учун буйруқлар сатрида ҳарфнинг номи киритилади. Масалан, α белгисини ҳосил қилиш учун **alpha** сўзини киритиш керак.

Грек алфавити ҳарфлари ва уларнинг номлари:

α - alpha	ν - nu
β - beta	μ - mu
γ - gamma	ξ -xi
δ - delta	π - pi
ε - epsilon	ρ - rho
ζ - zeta	σ - sigma
η - eta	υ - upsilon
θ - theta	ϕ - phi
ι - ita	χ - chi
κ - kappa	ψ - psi
λ - lambda	ω -omega

Грек алфавитини бош ҳарфларини киритиш учун, мос ҳарфнинг номини бош ҳарф билан киритиш керак масалан Ω ҳарфини ҳосил қилиш учун **Omega** тарзида киритиш керак. Грек алфавити ҳарфларини махсус палитралар ёрдамида ҳам киритиш мумкин.

Назорат саволлари

1. *Maple тизими нима учун мўлжаллангани?*
2. *Maple ойнасининг асосий элементларини тавсифланг.*
3. *Maple ишчи майдони неча қисмга бўлинади ва бу қисмларда қандай маълумотлар акс этади?*
4. *Қандай қилиб буйруқлар сатри матнли сатрга ўтказилади?*
5. *Maple да асосий меню бандларини тушунтириб беринг.*
6. *Ишчи варақ файлининг стандарт кенгайтмаси қандай?*
7. *Maple да асосий математик ўзгармаслар қандай тасвирланади?*
8. *Рационал сонларнинг берилиш усуларини тавсифланг.*
9. *Рационал сонларнинг тақрибий қийматлари қандай ҳосил қилинади?*

10. Maple командалари қанақа белгилар билан тугайди ва улар бир-биридан нимаси билан фарқ қилади?
11. Қисм дастурлар кутубхонаси қайси команда орқали чақирилади?

2-Мавзу: Maple тизими командасининг синтаксиси. Математик ифодаларни қайта ишлаш командалари

Режа:

1. Команда синтаксиси. Стандарт функциялар
2. Математик шакл алмаштиришлар
3. Функцияларнинг бериш усуллари. Ўзгарувчиларни алмаштириш
4. Баҳолаш операцияси

1. Команда синтаксиси. Стандарт функциялар

Команда синтаксиси. *Maple* тизими командаси буйруқ номи ва параметрларидан иборат: **command(p1, p2, ...)**. Ҳар бир команда (;) ёки (:) билан тугалланади. Команда (;) белгиси билан тугаса команда бажарилишининг натижаси чиқариш майдонида ҳосил бўлади, (:) белгиси билан тугаса команда бажарилади, лекин унинг натижаси чиқариш майдонида ҳосил бўлмайди.

(%) белгиси олдинги команданинг натижасини билдиради. (%) белгисидан фойдаланишни қуйидаги мисолда кўрамиз:

> a+b;

$a+b$

> %+c;

$a+b+c.$

Ўзгарувчига қиймат таъминлаш учун (:=) таъминлаш белгиси ишлатилади.

Maple программаси ишга туширилганда унинг барча командалари тўлиқлигича хотирада юкланмаган бўлади. Командаларнинг кўп қисми топиш кўрсаткичларига эга бўлади ва чақирув жараёнида автоматик равишда ишга тушади. Айрим командалар стандарт кутубхоналарда бўлади уларни бажаришдан аввал албатта **readlib(command)** командасини бажариш керак, бу ерда **command** — чақирилаётган команда номи. *Maple* процедураларининг

қолган қисми пакет деб аталувчи махсус кутубхоналарда сақланади.

Пакетларни чақиришнинг икки хил усули мавжуд:

1) **with(package)** командаси ёрдамида бутун пакетни чақириш мумкин бу ерда **package** – пакет номи;

2) **package[command](options);** махсус форматдаги команда орқали биринчи пакет номи **package**, квадрат қавс ичида керакли команда **command**, ва қавс ичида команданинг параметрлари **options** терилади.

Maple программаси кутубхоналарига қуйидаги пакетлар киради: **linalg** – чизиқли алгебра амалларини ўз ичига олади; **geometry** – планиметрия масалаларини ечиш учун; **geom3d** – стереометрия масалаларини ечиш учун; **student** – аналитик кўринишда ечилган масалаларни ечиш босқичларини ўз ичига олади.

Maple тизимида қуйидаги стандарт функциялар мавжуд:

Математик ифодаси

Maple даги ифодаси

e^x

exp(x)

$\ln x$

ln(x)

$\lg x$

log10(x)

$\log_a x$

log[a](x)

$\sqrt{x}$

sqrt(x)

$|x|$

abs(x)

$\sin x$

sin(x)

$\cos x$

cos(x)

$\operatorname{tg} x$

tan(x)

$\operatorname{ctg} x$

cot(x)

$\sec x$

sec(x)

$\operatorname{cosec} x$

csc(x)

$\arcsin x$

arcsin(x)

$\arccos x$

arccos(x)

$\arctg x$	$\arctan(x)$
$\arcctg x$	$\operatorname{arccot}(x)$
$\operatorname{sh} x$	$\sinh(x)$
$\operatorname{ch} x$	$\cosh(x)$
$\operatorname{th} x$	$\tanh(x)$
$\operatorname{cth} x$	$\coth(x)$
$\delta(x)$ - Дирак функцияси	$\operatorname{Dirac}(x)$
$\theta(x)$ - Хевиссайд функцияси	$\operatorname{Heaviside}(x)$

Maple да юқоридаги функциялардан ташқари Бессел функцияси, Эйлер бета- ва гамма– функциялари, интеграл хато, эллиптик интеграллар, турли хил ортогонал кўпхадлар каби кўп сонли функциялар мавжуд.

2. Математик шакл алмаштиришлар

Maple математик формулаларни аналитик шакл алмаштиришда кенг имкониятга эга. Буларга ўхшаш ҳадларни ихчамлаш, кўпайтувчиларга ажратиш, кавсларни очиш касрни иррационалликдан кутқариш каби амаллар киради.

Ифодани қисмларга ажратиш. Шакл алмаштири-ладиган ифода қуйидаги кўринишда ёзилади: **eq:=exp1=exp2**; бу ерда **eq** – ихтиёрий ифода номи, **exp1** – ифоданинг чап қисми, **exp2** – ифоданинг ўнг қисми.

Ифоданинг ўнг қисмини ажратиш **rhs(eq)** командаси орқали, ифоданинг Нчап қисмини ажратиш эса **lhs(eq)** командаси орқали амалга оширилади. Мисол:

> eq:=a^2-b^2=c;

$$eq := a^2 - b^2 = c$$

> lhs(eq);

$$a^2 - b^2$$

> rhs(eq);

c

Агар ифода a/b рационал каср кўринишида берилган бўлса, унинг сурат ва махражини мос равишда **numer** ва **denom** командалари орқали ажратиш мумкин. Мисол:

> f:=(a^2+b)/(2*a-b);

$$f := \frac{a^2 + b}{2a - b}$$

> numer(f);

$$a^2 + b$$

> denom(f);

$$2a - b$$

Ифодаларни айнан шакл алмаштириш. Қавсларни очиш **expand(eq)** командаси орқали амалга оширилади. Мисол:

> eq:=(x+1)*(x-1)*(x^2-x+1)*(x^2+x+1);

$$eq := (x + 1)(x - 1)(x^2 - x + 1)(x^2 + x + 1)$$

> expand(eq);

$$x^6 - 1$$

Кўпхадни кўпайтувчиларга ажратиш **factor(eq)** командаси орқали амалга оширилади. Мисол:

> p:=x^5-x^4-7*x^3+x^2+6*x;

$$p := x^5 - x^4 - 7x^3 + x^2 + 6x$$

> factor(p);

$$x(x - 1)(x - 3)(x + 2)(x + 1)$$

expand командаси қўшимча параметрларга эга бўлиши мумкин, яъни қавсларни очиш жараёни маълум бир ифодага нисбатан ўзгаришсиз қолдирилади. Масалан, $\ln x + e^x - y^2$ ифоданинг ҳар бир ҳадини $(x+a)$ ифодага кўпайтириш талаб қилинсин, у ҳолда буйруқлар сатрида:

```
> expand((x+a)*(ln(x)+exp(x)-y^2), (x+a));
```

$$(x+a)\ln x + (x+a)e^x - (x+a)y^2$$

Каср кўринишидаги ифода **normal(eq)** командаси орқали нормал кўринишга келтирилади. Масалан:

```
> f:=(a^4-b^4)/((a^2+b^2)*a*b);
```

$$f := \frac{a^4 - b^4}{(a^2 + b^2)ab}$$

```
> normal(f);
```

$$\frac{a^2 - b^2}{ab}$$

Ифодани соддалаштириш **simplify(eq)** командаси орқали амалга оширилади. Масалан:

```
> eq:=(cos(x)-sin(x))*(cos(x)+sin(x));
```

```
> simplify(eq);
```

$$2\cos(x)^2 - 1$$

Ўхшаш ҳадларни ихчамлаш **collect(exp,var)** командаси орқали амалга оширилади. Бу ерда **exp** – ифода, **var** – ўхшашларини танлаш керак бўлган ўзгарувчи номи. **simplify** командасида параметр сифатида ифода турини олиш мумкин. Масалан, команда **simplify(eq,trig)** кўринишида берилганда тригонометрик муносабатларга нисбатан соддалаштирилади. Команданинг айнан бир функциялар синфига нисбатан соддалаштриш вариантлари: **power** – даражали ифодага нисбатан; **radical** ёки **sqrt** – илдизли ифодага нисбатан; **exp** – экспоненциал ифодага нисбатан; **ln** –логарифмик ифодага нисбатан. Параметрларни киритиш **simplify** командасини самарадорлигини оширади.

Ўхшаш ҳадларни бирлаштириш ёки тригонометрик функцияларни даражасини пасайтириш **combine(eq,param)** командаси орқали амалга оширилади, бу ерда **eq** – ифода, **param** – бирлаштириш қоидаларини кўрсатадиган параметр. Масалан

```
> combine(4*sin(x)^3, trig);
```

$$-\sin(3x) + 3\sin(x)$$

Квадрат илдиздан фарқли даражали илдизли ифодаларни соддалаштириш учун **radnormal(eq)** командасидан фойдаланилади. Масалан:

```
> sqrt(3+sqrt(3)+(10+6*sqrt(3))^(1/3))=
radnormal(sqrt(3+sqrt(3)+(10+6*sqrt(3))^(1/3)));
```

$$\sqrt{3 + \sqrt{3} + (10 + 6\sqrt{3})^{1/3}} = 1 + \sqrt{3}$$

convert(exp, param) командаси орқали **exp** – туридаги ифода **param** – туридаги ифодага алмаштирилади. Хусусан, $\sin x$ ва $\cos x$ қатнашган ифодаларни фақат $\tan x$ қатнашган ифодага **tan** параметрини киритиш орқали алмаштириш мумкин ва аксинча $\tan x$, $\cot x$ қатнашган ифодаларни **sincos** параметрини киритиш орқали $\sin x$ ва $\cos x$ қатнашган ифодаларга алмаштириш мумкин.

convert ифодани бир турдан бошқасига ўтказиш имкониятига ҳам эга, масалан: **convert(list, vector)** – командаси **list** рўйхат кўринишидаги ифодани шу элементлардан иборат вектор билан алмаштиради; **convert(expr, string)** – математик ифодани унинг матнли ёзувига алмаштиради. **convert** командасининг параметрлари ҳақида маълумотга эга бўлиш учун **convert[termin]** маълумотлар тизимида мурожаат қилиш керак бўлади.

Бирор команданинг параметрлари ёки унинг қўлланиши ҳақида маълумотга эга бўлиш учун *Maple* нинг маълумотлар тизимидан фойдаланиш мумкин. Бунинг учун керакли командани белгилаб F1 тугмасини босиш керак, агар команда тўғри терилган бўлса, команда тавсифи ҳосил бўлади.

3. Функцияларни бериш усуллари. Ўзгарувчиларни алмаштириш

Maple да Функцияни тавсифлашнинг бир неча усуллари мавжуд.

1- усул. Таминлаш оператори (**:=**) ёрдамида функцияни тавсифлаш.

Масалан:

```
> f:=sin(x)+cos(x);
```

$$f := \sin(x) + \cos(x)$$

Агар x ўзгарувчига аниқ бир қиймат берилган бўлса, у ҳолда f функция қиймати x учун ҳисобланади. Масалан, юқоридаги мисолни давом эттирсак ва f нинг қийматини $x = \pi / 4$ бўлганда ҳисоблаш учун қуйидагича ёзиш керак:

```
> x:=Pi/4;
```

$$x := \frac{\pi}{4}$$

```
> f;
```

$$\sqrt{2}$$

Бу буйруқлардан кейин x ўзгарувчи $\pi / 4$ қийматга эга бўлади. Ўзгарувчига аниқ қийматни тўлиқ таъминламаслик учун **subs({x1=a1, x2=a2,...},f)** командасидан фойдаланиш қулай, фигурали қавс ичида f функцияга қўйилиши керак бўлган x_i ўзгарувчилар ва уларнинг янги қийматлари a_i ($i=1,2,\dots$) кўрсатилади. Масалан:

```
> f:=x*exp(-t);
```

$$f := xe^{(-t)}$$

```
> subs({x=2,t=1},f);
```

$$2e^{(-1)}$$

Maple да жимлик қондаси бўйича барча ҳисоблашлар символли бажарилади, яъни натижа ўзида e, π каби ирроционал ўзгармаслардан иборат ифода кўринишида чиқарилади. Тақрибий қийматни кўзгалувчи нуқтали кўринишда олиш учун **evalf(expr,t)** командасидан фойдаланамиз, бу ерда **expr** – ифода, **t** – ифоданинг нуқтадан кейинги аниқлиги. Масалан, юқоридаги мисолни давом эттирсак функциянинг тақрибий қийматини ҳисоблаймиз:

```
> evalf(%);
```

$$.7357588824$$

Бу ерда (%) белги ўзидан юқоридаги буйруқ натижасини қабул қилади.

2-ўсул. Функционал оператор ёрдамидан функцияни аниқлаш, яъни ўзгарувчилар тўплами $(x_1, x_2, \dots)$ ва бир ёки бир нечта ифодалар $(f_1, f_2, \dots)$

> **f:=piecewise(x<0, 0, 0<=x and x<1, x, x>=1, sin(x));**

$$f := \begin{cases} 0 & x < 0 \\ x & -x \leq 0 \text{ and } x - 1 < 0 \\ \sin x & 1 \leq x \end{cases}$$

4. Баҳолаш операцияси

Ҳақиқий ифодаларни баҳолаш. *Maple* да ҳақиқий ифодаларни баҳолайдиган қуйдаги командалар мавжуд:

frac(expr) – **expr** ифоданинг каср қисмини ҳисоблаш;

trunc(expr) – **expr** ифоданинг бутун қисмини ҳисоблаш;

round(expr) – **expr** ифодани яхлитлаш;

Комплекс ифодаларни баҳолаш. $z=x+iy$ комплекс ифоданинг ҳақиқий ва мавҳум қисмини қийматлари **Re(z)** ва **Im(z)** командалари ёрдамида топилади. Масалан:

> **z:=3+I*2;**

> **Re(z);Im(z);**

3, 2

Агар $z=x+iy$, бўлса у ҳолда унга комплекс тескари $w=z^*=x-iy$ ифодани **conjugate(z)** командаси орқали топиш мумкин. Юқоридаги мисолни давом эттирсак:

> **w:=conjugate(z);**

$w:=3-2 I$

z комплекс ифоданинг модули ва аргументини **polar(z)** командаси ёрдамида топиш мумкин. Бу командадан фойдаланишдан олдин уни стандарт кутубхонадан **readlib** командаси билан чақириб олиш керак. Масалан:

> **readlib(polar); polar(I);**

$$\text{polar}\left(1, \frac{1}{2}\pi\right)$$

қавс ичида i соннинг модули (1) ва унинг аргументи ($\pi/2$) вергул билан ажратилган.

Агар комплекс сон жуда мураккаб ёки параметрлар билан берилган бўлса, **Re(z)** ва **Im(z)** командалар талаб қилинган натижани бермайди. **z** ифоданинг ҳақиқий ва мавҳум қисмини комплекс ифодаларни аниқловчи **evalc(z)** командаси орқали топиш мумкин. Масалан:

> z:=ln(1-I*sqrt(3))^2;

$$z := \ln(1 - I\sqrt{3})^2$$

> evalc(Re(z)); evalc(Im(z));

$$\frac{1}{4} \ln(4)^2 - \frac{1}{9} \pi^2$$

$$- \frac{1}{3} \ln(4) \pi$$

Назорат саволлари

1. Ифоданинг ўнг ва чап қисмини, касрнинг сурат ва махражсини ажратувчи командаларни тавсифланг
2. *factor*, *expand*, *normal*, *simplify*, *combine*, *convert* командаларининг вазифаларини тушунтиринг
3. Maple да функцияларни тавсифлашнинг асосий усулларини айтинг.
4. Maple да ҳақиқий ифодалар устида қандай баҳолаш амаллари бажариши мумкин?
5. *evalf* буйруғи қандай вазифани бажаради?
6. Қайси буйруқлар ёрдамида комплекс ифоданинг ҳақиқий ва мавҳум қисмини, унинг модули ва аргументини, унга комплекс тескари ифодани топиш мумкин?
7. *evalc* буйруғи қандай вазифани бажаради?

3-мавзу: Тенглама ва тенгсизликларни ечиш командалари

Режа:

1. Тенгламаларни ечиш
2. Тенгламалар системасини ечиш
3. Тенгсизликларни ечиш

1. Тенгламаларни ечиш

Оддий тенгламаларни ечиш. *Maple* да тенгламаларни ечиш учун универсал команда **solve(eq,x)** мавжуд, бу ерда **eq** – тенглама, **x** – қайси ўзгарувчига нисбатан тенгламани ечиш кераклигини кўрсатади. Бу команда бажарилиши натижасида чиқариш майдонида тенглама ечими ҳосил бўлади.

Масалан:

```
> solve(a*x+b=c,x);
```

$$-\frac{b-c}{a}$$

Агар тенглама бир нечта ечимга эга бўлиб, бу ечимлар кейинчалик керак бўлса **solve** командасини бирор бир ном **name** га таминлаш керак. Қандайдир **k**–ечимга мурожат қилиш учун, унинг номи ва ечим рақами **k** ни тўртбурчак қавс ичига ёзамиз: **name[k]**. Масалан:

```
> x:=solve(x^2-a=0,x);
```

$$x := -\sqrt{a}, \sqrt{a}$$

```
> x[1];
```

$$-\sqrt{a}$$

```
> x[2];
```

$$\sqrt{a}$$

```
> x[1]+x[2];
```

$$0$$

Тенгламаларни сонли ечиш. Трансцендент тенгламалар аналитик ечимга эга бўлмаганида уларнинг сонли ечимини махсус **fsolve(eq,x)**

командаси ёрдамида топилади. Бу команданинг параметрлари **solve** командасиникидек. Масалан:

```
> x:=fsolve(cos(x)=x,x);
```

$x:=.7390851332$

Рекуррент ва функционал тенгламаларни ечиш.

rsolve(eq, f) командаси **eq** рекуррент тенгламани **f** бутун функцияси учун ечишга имкон беради. **f(n)** функцияси учун бошланғич шарт бериш мумкин бунда мазкур рекуррент тенгламанинг хусусий ечимига эга бўламиз. Масалан:

```
> eq:=2*f(n)=3*f(n-1)-f(n-2);
```

$$eq := 2f(n) = 3f(n-1) - f(n-2)$$

```
> rsolve({eq,f(1)=0,f(2)=1},f);
```

$$2 - 4\left(\frac{1}{2}\right)^n$$

Универсал **solve** командаси функционал тенгламаларни ечиш имконини беради. Масалан:

```
> F:=solve(f(x)^2-3*f(x)+2*x,f);
```

$F := \text{proc}(x) \text{ RootOf}(_Z^2 - 3_Z + 2*x) \text{ end}$

Натижада ечим ошкормас кўринишда ҳосил бўлади. Аммо *Maple* бундай ечимлар билан ишлай олади. Функционал тенгламанинг ошкормас ечимини **convert** командаси ёрдамида бирор бир элементар функцияга айлантириш мумкин. Юқоридаги мисолни давом этирган ҳолда натижани ошкор кўринишда олиш мумкин:

```
> f:=convert(F(x),radical);
```

$$f := \frac{3}{2} + \frac{1}{2}\sqrt{9-8x}$$

Тригонометрик тенгламаларни ечиш. **solve** командасини тригонометрик тенгламани ечишда қўллаганда унинг фақат хусусий ечимини

беради, яъни $[0, 2\pi]$ оралиғидаги ечимларини беради. Умумий ечимни олиш учун олдиндан қўшимча **_EnvAllSolutions:=true** командасини киритамиз. Масалан:

```
> _EnvAllSolutions:=true;
> solve(sin(x)=cos(x),x);
```

$$\frac{1}{4}\pi + \pi_Z \sim$$

Maple да $_Z \sim$ символ бутун турдаги константани ифодалайди, шунинг учун мазкур мисолнинг одатдаги кўриниши қуйдагича $x := \pi / 4 + \pi n$, бу ерда n – бутун сонлар.

Трансцендент тенгламани ечиш. Трансцендент тенгламаларни ечишда ечимни ошкор кўринишда олиш учун **solve** командасидан олдин **_EnvExplicit:=true** командасини киритиш керак. Қуйидаги мисолда трансцендент тенгламаларнинг мураккаб системасининг ечими ва натижани соддалаштириш кўрсатилган:

```
> eq:={ 7*3^x-3*2^(z+y-x+2)=15, 2*3^(x+1)+
3*2^(z+y-x)=66, ln(x+y+z)-3*ln(x)-ln(y*z)=-ln(4) };
> _EnvExplicit:=true;
> s:=solve(eq,{x,y,z});
> simplify(s[1]);simplify(s[2]);
{x=2, y=3, z=1}, {x=2, y=1, z=3}
```

2. Тенгламалар системасини ечиш.

Тенгламалар системаси ҳам **solve({eq1,eq2,...},{x1,x2,...})** командаси ёрдамида ечилади, фақат команда параметрларида, биринчи фигурали қавс ичида тенгламаларни вергул билан ажратиб ёзиб чиқамиз, иккинчисида эса ўзгарувчилар рўйхати вергул билан ажратилиб ёзилади. Агар ечимлар кейинчалик керак бўлса **solve** командасини бирор бир ном **name** га таъминлаш керак. Сўнгра **assign(name)** командаси бажарилади. Шундан кейингина

ечимлар устида бирор бир математик амалларни бажариш мумкин бўлади.

Масалан:

```
> s:=solve({a*x-y=1,5*x+a*y=1},{x,y});
```

$$s:=\left\{x=\frac{a+1}{5+a^2}, y=\frac{a-5}{5+a^2}\right\}$$

```
> assign(s); simplify(x-y);
```

$$\frac{6}{5+a^2}$$

3. Тенгсизликларни ечиш

Оддий тенгсизликларни ечиш. **solve** командаси тенгсизликни ечишда ҳам ишлатилади. Тенгсизликнинг ечими мазкур ўзгарувчининг ўзгариш оралиғи кўринишида чиқади. Агар тенгсизликнинг ечими ярим ўқ бўлса, у ҳолда чиқариш майдонида $\text{RealRange}(-\infty, \text{Open}(a))$ кўринишдаги конструкция чиқади, қайсиким $x \in (-\infty, a)$ ва a - бирор бир сон. Open сўзи интервал очик чегарали деган маънони беради. Агар бу сўз бўлмаса, у ҳолда интервал чегараси ечимлар тўпламига киришини билдиради. Масалан:

```
> s:=solve(sqrt(x+3)<sqrt(x-1)+sqrt(x-2),x):
```

```
> convert(s,radical);
```

$$\text{RealRange}\left(\text{Open}\left(\frac{2}{3}\sqrt{21}\right), \infty\right)$$

Агар тенгсизликнинг ечимини $x \in (a, b)$ оралиқ тўплам кўринишида эмас, $a < x$, $x < b$ тенгсизликлар кўринишида олиш учун тенгсизлик қайси ўзгарувчига нисбатан ечилаётган бўлса ўша ўзгарувчини фигурали қавс ичига ёзиш керак.

Масалан:

```
> solve(1-1/2*ln(x)>2,{x});
```

$$\{0 < x, x < e^{(-2)}\}$$

Тенгсизликлар системасини ечиш. **solve** командаси ёрдамида тенгсизликлар системасини ҳам ечиш мумкин. Масалан:

> solve({x+y>=2,x-2*y<=1,x-y>=0,x-2*y>=1},{x,y});

$$\{x = 1 + 2y, \frac{1}{3} \leq y\}$$

Энди $13x^3 - 25x^2 - x^4 - 129x + 270 > 0$ тенгсизликни ечиш жарёнини кўрамиз:

> solve(13*x^3-25*x^2-x^4-129*x+270>0,x);

RealRange(Open(-3), Open(2)), RealRange(Open(5), Open(9))

Назорат саволлари

1. *solve* командасининг вазифаси.
2. Қайси командалар тенгламанинг сонли ечимини ва рекуррент тенгламаларни ечимини топшида қўлланилади?
3. Тенгламанинг умий ечимини ва хусусий ечимини олиш учун қандай қўшимча буйруқлар берилади?
4. Тенгсизликнинг ечими қай шаклда чиқарилади?
5. Чиқариш қаторида ёпиқ интервални очиқ интервалдан қандай ажратиш мумкин?
6. Тенгламалар системасини ечиш учун *solve* командасига қандай қўшимча параметрлар киритилади?
7. Тригонометрик тенгламаларнинг умий ечимини олиш учун *solve* командасидан олдин қайси команда киритилиши керак?
8. Трансцендент тенламални ечиш учун *solve* командасидан олдин қайси команда киритилиши керак?

4- мавзу: Чизиқли алгебра амалларини бажариш командалари

Режа:

1. Векторларнинг берилиш усуллари ва улар устида амаллар
2. Матрицалар билан ишлаш
3. Матрицаларнинг спектрал анализи
4. Чизиқли тенгламалар системаси. Матрицали тенгламалар

1. Векторларнинг берилиш усуллари ва улар устида амаллар

Чизиқли алгебра масалаларини ечиш командаларининг асосий қисми **linalg** кутубхонасида бўлганлиги сабабли уларни ечишдан аввал бу кутубхонани **with(linalg)** командаси орқали юклаб оламиз.

Векторларнинг берилиш усуллари. *Maple* да вектор ҳосил қилиш учун **vector([x1,x2,...,xn])** командасидан фойдаланилади, бунда векторнинг координаталари квадрат қавсда вергуллар билан ажратилади. Масалан:

```
> x:=vector([1,0,0]);
```

```
x:=[1, 0, 0]
```

Аниқланган x векторнинг координатасини чиқариш майдонида ҳосил қилиш учун **x[i]** командасидан фойдаланилади. Бу ерда i координата номери. Масалан, юқорида келтирилган векторнинг биринчи координатасини қуйидагича чиқариш мумкин:

```
> x[1];
```

1

Векторни рўйхатгага айлантириш ёки аксинча рўйхатни векторга айлантириш мумкин, бунинг учун қуйидаги командалардан фойдаланамиз: **convert(vector, list)** ёки **convert(list, vector)**.

Векторларни қўшиш. Иккита a ва b векторни қўшишни икки хил команда ёрдамида амалга ошириш мумкин:

1) **evalm(a+b)**

2) **matadd(a, b)**

a ва b векторларнинг чизиқли комбинациясини ҳисоблаш **add** командаси орқали амалга оширилади: $\alpha a + \beta b$, бу ерда α, β – скаляр катталиклар бўлиб, форматдан фойдалансак: **matadd(a, b, alpha, beta)**.

Векторларнинг скаляр ва вектор кўпайтмаси. Векторлар орасидаги бурчак. Икки векторнинг $(a, b) = \sum_{i=1}^n a_i b_i$ скаляр кўпайтмаси **dotprod(a, b)** командаси ёрдамида ҳисобланади.

Икки векторнинг $[a, b]$ вектор кўпайтмаси **crossprod(a, b)** командаси ёрдамида ҳисобланади.

a ва b векторлар орасидаги бурчак **angle(a, b)** командаси ёрдамида ҳисобланади.

Векторнинг нормаси. $a = (x_1, \dots, x_n)$ векторнинг $\|a\| = \sqrt{x_1^2 + \dots + x_n^2}$ нормаси (узунлиги) **norm(a, 2)** командаси орқали аниқланади.

a векторни **normalize(a)** командаси ёрдамида нормаллаштириш мумкин. Бу команда бажарилиши натижасида $\frac{a}{\|a\|}$ бирлик узунликка эга вектор ҳосил бўлади.

Векторлар системасининг базисини топиш. Грамм-Шмидт процедураси орқали векторлар системасини ортогоналлаштириш. n та вектордан иборат $\{a_1, a_2, \dots, a_n\}$ системанинг базиси **basis([a1, a2, ..., an])** командаси орқали аниқланади.

Чизиқли боғланмаган $\{a_1, a_2, \dots, a_n\}$ векторлар системаси **GramSchmidt([a1, a2, ..., an])** командаси ёрдамида ортогоналлаштирилади.

2. Матрицалар билан ишлаш

Матрицани ҳосил қилиш. *Maple* да матрица ҳосил қилиш учун **matrix(n, m, [[a11,a12,...,a1n], [a21,a22,...,a2m],..., [an1,an2,...,anm]])** командасидан фойдаланамиз. Бу ерда n – сатрлар сони, m – устунлар сони. Сатр ва устунлар сонини кўрсатмасдан, матрица элементларини сатрлар бўйича квадрат қавсда кетма-кет вергул орқали бериш ҳам мумкин. Масалан:

> A:=matrix([[1,2,3],[-3,-2,-1]]);

$$A := \begin{bmatrix} 1 & 2 & 3 \\ -3 & -2 & -1 \end{bmatrix}$$

Maple да махсус турдаги матрицаларни қўшимча командалар орқали ҳосил қилиш мумкин. Хусусий ҳолда диоганал матрица ҳосил қилиш учун **diag** командасидан фойдаланамиз. Масалан:

> J:=diag(1,2,3);

$$J := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Матрицани i ва j ўзгаувчиларга боғлиқ $f(i, j)$ функция орқали ҳосил қилиш мумкин, бу ерда i, j – матрица индекслари. Бунинг учун **matrix(n, m, f)** командасидан фойдаланилади. Бу ерда n – сатрлар сони, m – устунлар сони. Масалан:

> f:=(i, j)->x^i*y^j;

$$f := (i, j) \rightarrow x^i y^j$$

> A:=matrix(2,3,f);

$$A := \begin{bmatrix} xy & xy^2 & xy^3 \\ x^2y & x^2y^2 & x^2y^3 \end{bmatrix}$$

А матрицанинг сатрлари сони **rowdim(A)**, устунлари сони **coldim(A)** буйруғи орқали аниқланади.

Матрицалар устида арифметик амаллар. Ўлчамлари бир хил бўлган икки матрицани қўшиш учун, иккита векторни қўшиш каби **evalm(A+B)** ёки

matadd(A, B) буйруқларидан фойдаланилади. Иккита матрицанинг кўпайтмаси куйидаги буйруқлар ёрдамида топилади:

1) **evalm(A&*B)**

2) **multiply(A, B)**

Матрицани кўпайтириш командаларида иккинчи аргумент сифатида векторни ҳам кўрсатиш мумкин, масалан:

```
> A:=matrix([[1,0],[0,-1]]);
```

```
> B:=matrix([[ -5,1], [7,4]]);
```

$$A := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad B := \begin{bmatrix} -5 & 1 \\ 7 & 4 \end{bmatrix}$$

```
> v:=vector([2,4]);
```

$$v := [2, 4]$$

```
> multiply(A,v);
```

$$[2, -4]$$

```
> multiply(A,B);
```

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

```
> matadd(A,B);
```

$$\begin{bmatrix} -4 & 1 \\ 7 & 3 \end{bmatrix}$$

evalm командаси матрицани сонга кўпайтириш ва матрицага сонни қўшиш имконини беради. Масалан:

```
> C:=matrix([[1,1],[2,3]]);
```

```
> evalm(2+3*C);
```

$$\begin{bmatrix} 5 & 3 \\ 6 & 11 \end{bmatrix}$$

Матрицанинг детерминанти, минори и алгебраик тўлдирувчиси.

Матрицанинг ранги ва изи. A матрицанинг детерминанти **det(A)** командаси орқали топилади.

minor(A, i, j) командаси A матрицанинг i -сатри ва j -устунини ўчиришдан ҳосил бўлган минорини аниқлайди. A матрица a_{ij} элементининг M_{ij} минори **det(minor(A,i,j))** командаси ёрдамида аниқланади. A матрицанинг ранги **rank(A)** командаси орқали топилади. A матрицанинг изи унинг диагональ элементлари йиғиндисига тенг ва у **trace(A)** командаси орқали аниқланади.

```
> A:=matrix([[4,0,5],[0,1,-6],[3,0,4]]);
```

$$A := \begin{bmatrix} 4 & 0 & 5 \\ 0 & 1 & -6 \\ 3 & 0 & 4 \end{bmatrix}$$

```
> det(A);
```

1

```
> minor(A,3,2);
```

$$\begin{bmatrix} 4 & 5 \\ 0 & -6 \end{bmatrix}$$

```
> det(%);
```

-24

```
> trace(A);
```

9

Тескари ва транспонирланган матрица. $A^{-1}A=AA^{-1}=E$ шартни қаноатлантирувчи A^{-1} тескари матрица икки хил усул билан аниқланади. Бу ерда E – бирлик матрица:

1) **evalm(1/A)**

2) **inverse(A)**

A' - транспонирланган матрица A – матрица сатрларини устунлари билан алмаштиришдан ҳосил бўлади. Транспонирланган матрицани топиш учун **transpose(A)** командасидан фойдаланилади.

Масалан, юқорида берилган A матрица учун тескари ва транспонирланган матрицани топамиз:

> inverse(A);

$$\begin{bmatrix} 4 & 0 & -5 \\ -18 & 1 & 24 \\ -3 & 0 & 4 \end{bmatrix}$$

> multiply(A, %);

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> transpose(A);

$$\begin{bmatrix} 4 & 0 & 3 \\ 0 & 1 & 0 \\ 5 & -6 & 4 \end{bmatrix}$$

Матрицанинг турини аниқлаш. Матрицанинг мусбат ёки манфий аниқланганлиги **definite(A, param)** командаси ёрдамида аниқланади. Бу ерда **param** қуйидаги қийматларни олиши мумкин: ' **positive_def** ' – мусбат аниқланган ($A > 0$), ' **positive_semidef** ' – манфиймас аниқланган ($A \geq 0$), ' **negative_def** ' – манфий аниқланган ($A < 0$), ' **negative_semidef** ' – мусбатмас аниқланган ($A \leq 0$). Натижа мантикий ўзгармаслар билан берилади: **true** – чин, **false** – ёлгон. Масалан:

> A:=matrix([[2,1],[1,3]]);

$$A := \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

> definite(A,'positive_def');

true

A матрицанинг ортогоналлигини **orthog(A)** командаси ёрдамида текшириш мумкин.

> B:=matrix([[1/2,1*sqrt(3)/2], [1*sqrt(3)/2,-1/2]]);

$$B := \begin{bmatrix} \frac{1}{2} & \frac{1}{2}\sqrt{3} \\ \frac{1}{2}\sqrt{3} & -\frac{1}{2} \end{bmatrix}$$

> orthog(B);

true

Матрицага боғлиқ функциялар. A матрицани n даражага кўтариш **evalm(A^n)** командаси орқали бажарилади. e^A ни ҳисоблаш учун **exponential(A)** командадан фойдаланилади. Масалан:

> T:=matrix([[5*a,2*b],[-2*b,5*a]]);

$$T := \begin{bmatrix} 5a & 2b \\ -2b & 5a \end{bmatrix}$$

> exponential(T);

$$\begin{bmatrix} e^{(5a)} \cos(2b) & e^{(5a)} \sin(2b) \\ -e^{(5a)} \sin(2b) & e^{(5a)} \cos(2b) \end{bmatrix}$$

> evalm(T^2);

$$\begin{bmatrix} 25a^2 - 4b^2 & 20ab \\ -20ab & 25a^2 - 4b^2 \end{bmatrix}$$

3. Матрицаларнинг спектрал анализи

Матрицаларнинг хос сон ва хос векторлари. Чизикли алгебра курсидан маълумки агар $Ax = \lambda x$ бўлса у ҳолда x вектор A матрицанинг хос вектори, λ сон эса матрицанинг x векторга мос келадиган хос сони дейилади. Матрица барча хос сонларининг мажмуаси матрицанинг спектри дейилади. Агар матрица спектрида битта хос сон k марта такрорланса у ҳолда бу хос соннинг карралилиги k га тенг дейилади.

A матрицанинг хос сонини топиш учун **eigenvalues(A)** командасидан, хос векторини топиш учун **eigenvectors(A)** командасидан фойдаланилади. Бу

команданинг бажарилиши натижасида хос сонлар, уларнинг карралилиги ва мос хос векторлар ҳосил бўлади.

eigenvectors команданинг бажарилишидан ҳосил бўлган натижа қандай кўринишда ҳосил бўлишини тушунинш учун қуйидаги мисолни келтирамиз.

$A = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 5 & -1 \\ 1 & -1 & 3 \end{bmatrix}$ матрица берилган бўлсин. Бу матрица 3 та хос векторга

эга: карралилиги 1 га тенг бўлган $\lambda_1 = 2$ хос сонга мос келадиган $a_1 = (-1, 0, 1)$ хос вектор, карралилиги 1 га тенг бўлган $\lambda_2 = 3$ хос сонга мос келадиган $a_2 = (1, 1, 1)$ хос вектор, карралилиги 1 га тенг бўлган $\lambda_3 = 6$ хос сонга мос келадиган $a_3 = (1, -2, 1)$ хос вектор. Буларни *Maple* да топамиз:

```
> A:=matrix([[3,-1,1],[-1,5,-1],[1,-1,3]]):
```

```
> eigenvectors(A);
```

```
[2,1,{{-1,0,1}}], [3,1,{{1,1,1}}], [6,1,{{1,-2,1}}]
```

Матрицанинг характеристик ва минимал кўпхад-лари. A матрицанинг $P_A(\lambda) = \det(\lambda E - A)$ характеристик кўпхадини ҳисоблаш учун **charpoly(A, lambda)** командасидан фойдаланилади.

A матрицанинг минимал кўпхад (бўлувчиси) **minpoly(A, lambda)** командаси ёрдамида топилади

Матрицаларнинг каноник ва махсус турлари. A матрицани Жордан нормал шаклига келтириш учун **jordan(A)** командасидан фойдаланилади.

A матрицани қуйидаги командалар ёрдамида учбурчак кўринишига келтириш мумкин:

1) **gausselim(A)** командаси A матрицани Гаусс усули ёрдамида учбурчак кўринишига келтиради;

Матрица ядроси. A матрица ядроси – бу шундай x векторлар тўпламики, уларнинг A матрица билан кўпайтмаси нол векторга тенг: $Ax = 0$. A матрица ядросини топиш бир жинсли чизиқли тенгламалар системасини ечимини топишга эквивалент. A матрица ядросини **kernel(A)** командаси ёрдамида топиш мумкин.

Назорат саволлари

1. Чизиқли алгебра масалаларини ечишдан олдин қайси пакетни юклаш керак?
2. Вектор ва матрицалар қайси командалар ёрдамида киритилади?
3. Қайси командалар ёрдамида иккита бир ўлчамдаги векторларни(матрицаларни) қўйиш мумкин?
4. Векторлар устида қандай турдаги кўпайтмалар бажарилади ва бунинг учун қайси командалардан фойдаланилади?
5. Вектор нормаси ва векторлар орасидаги бурчак қандай ҳисобланади?
6. Матрица детерминанти, минорлари, алгебраик тўлдирувчиси ва изини топиш командаларини тавсифланг.
7. Тескари ва транспонирланган матрицани топиш командаларини келтиринг.
8. Матрицанинг хос сон ва хос векторлари қандай топилади?
9. Чизиқли тенгламалар системасини ечиш командасини тавсифланг.
10. Матрицали тенламаларни ечиш учун қайси командадан фойдаланилади?

5-мавзу: Дифференциаллаш ва интеграллаш командалари

Режа:

1. Дифференциаллаш командалари

2. Интеграллаш командалари

1. Дифференциаллаш командалари

Ҳосилани ҳисоблаш. *Maple* да ҳосилани ҳисобловчи иккити команда мавжуд:

1) Тўғридан тўғри бажарилувчи **diff(f,x)** командаси, бу ерда **f**—функция **x**-ўзгарувчига нисбатан дифференциал-ланувчи функция.

2) **Diff(f,x)**-командаси эса ҳосиланинг $\frac{\partial}{\partial x} f(x)$ кўринишидаги математик талқинини ҳосил қилади. Ҳосила ҳисоблангач уни иложи борида соддалаштириш керак бўлади бунинг учун **simplify**, **factor** ёки **expand** командаларидан фойдаланамиз.

Мисол:

```
> Diff(sin(x^2),x)=diff(sin(x^2),x);
```

$$\frac{\partial}{\partial x} \sin(x^2) = 2 \cos(x^2) x$$

Юқори тартибли ҳосилаларни ҳисоблаш учун **x\$n** параметрни киритамиз, бу ерда **n** – ҳосиланинг тартиби. Мисол:

```
> Diff(cos(2*x)^2,x$4)=diff(cos(2*x)^2,x$4);
```

$$\frac{\partial^4}{\partial x^4} \cos(2x)^2 = -128 \sin(2x)^2 + 128 \cos(2x)^2$$

Олинган натижани икки хил усул билан соддалаштириш мумкин:

```
> simplify(%);
```

$$\frac{\partial^4}{\partial x^4} \cos(2x)^2 = 256 \cos(2x)^2 - 128$$

```
> combine(%);
```

$$\frac{\partial^4}{\partial x^4} \left(\frac{1}{2} \cos(4x) + \frac{1}{2} \right)^2 = 128 \cos(4x)$$

Дифференциал оператор. Дифференциал операторни аниқлаш учун **D(f)** командаси ишлатилади. Бу ерда **f**-функция. Мисол:

> D(sin);

cos

Бирор нуқтада ҳосилани ҳисоблаш учун қуйидаги команда киритилади:

> D(sin)(Pi):eval(%);

-1

Дифференциаллаш оператори функционал операторларда ҳам қўлланилади

> f:=x-> ln(x^2)+exp(3*x):

> D(f);

$$x \rightarrow 2 \frac{1}{x} + 3e^{(3x)}$$

> f:=x-> ln(x^2)+exp(3*x):

> D(f);

$$x \rightarrow 2 \frac{1}{x} + 3e^{(3x)}$$

2. Интеграллаш командалари

Аналитик сонли интеграллаш. $\int f(x)dx$ аниқмас интегрални ҳисоблаш командасининг икки хил кўриниши мавжуд:

1) команданинг актив ҳолати – **int(f, x)**, бу ерда **f** –интегралланувчи функция;

2) команданинг пассив ҳолати – **Int(f, x)**. **Int** командаси интеграл ифодасининг математик кўринишини экранда ҳосил қилади.

$\int_a^b f(x)dx$ аниқ интегрални ҳисоблаш учун **int** ва **Int** командаларга

қўшимча интеграллаш чегаралари киритилади. Масалан :

> Int((1+cos(x))^2, x=0..Pi)=int((1+cos(x))^2, x=0..Pi);

$$\int_0^{\pi} (1 + \cos(x))^2 dx = \frac{3}{2} \pi$$

Агар интеграллаш командасига continuous опциясини киритсак **int(f, x, continuous)**, у ҳолда *Maple* интеграллаш соҳасида интеграл остидаги функциянинг барча мумкин бўлган узилишларини четлаб ўтади. Бу чегараланмаган функцияларнинг хосмас интегралларини ҳисоблаш имконини беради. Агар **int** командасининг параметрларида **x=0..+infinity** чегара киритилса, чегаралари чексиз бўлган хосмас интегрални ҳисоблайди.

Сонли интеграл **evalf(int(f, x=x1..x2), e)** командаси орқали ҳисобланади, бу ерда **e** – ҳисоблаш аниқлиги.

Параметрларга боғлиқ интеграллар. Параметрлар учун чегаралар.

Агар параметрга боғлиқ интеграл ҳисобланса унинг қиймати ўша параметрга

ёки бирор бир шартга боғлиқ бўлади. Мисол сифатида $\int_0^{+\infty} e^{-ax} dx$ интегралини

караймиз. Бизга математик анализ курсидан маълумки, $a>0$ бўлганда бу интеграл яқинлашувчи ва $a<0$ бўлганда эса узоқлашувчи бўлади. Агар интегрални тўғридан-тўғри ҳисобласак, у ҳолда:

> Int(exp(-a*x),x=0..+infinity)=

int(exp(-a*x),x=0..+infinity);

Definite integration: Can't determine if the integral is convergent.

Need to know the sign of --> a

Will now try indefinite integration and then take limits.

$$\int_0^{+\infty} e^{-ax} dx = \lim_{x \rightarrow \infty} -\frac{e^{(-ax)} - 1}{a}.$$

натижага эга бўламиз.

Бу усул билан параметрга боғлиқ интегрални ҳисоблаб бўлмайди. Интегрални ҳисоблашда ошкор аналитик натижани олиш учун параметрлар ҳақида кандайдир фараз қилиш керак, яъни уларга чеклинишлар қўйилади. Буни **assume(expr1)** командаси орқали амалга ошириш мумкин, бу ерда **expr1** – тенгсизлик. Қўшимча чекланишлар **additionally(expr2)** командаси ёрдамида киритилади, бу ерда **expr2** – иккинчи тенгсизлик.

Параметрга чекланиш қўйилгандан сўнг *Maple* унинг олдига ($\sim$) симболини ўрнатади, масалан **a** параметрга бирор чекланиш қўйилган бўлса, у $a \sim$ кўринишини олади.

a параметрга қўйилган чекланишларни **about(a)** командаси орқали чақириш мумкин. Мисол: a параметрга $a > -1$, $a \leq 3$ чекланишларни қўямиз:

```
> assume(a>-1); additionally(a<=3);
```

```
> about(a);
```

Originally a, renamed $a \sim$:

is assumed to be: RealRange(Open(-1),3)

Юқоридаги $\int_0^{+\infty} e^{-ax} dx$ интегрални ҳисоблашга қайтамиз:

```
> assume(a>0);
```

```
> Int(exp(-a*x),x=0..+infinity)=int(exp(-*x),x=0..+infinity);
```

$$\int_0^{+\infty} e^{(-a \sim x)} dx = \frac{1}{a \sim}$$

Интеграллашнинг асосий усуллари. *Maple* да **student** пакети мавжуд бўлиб, у математикани ўқитишга мўлжалланган. У бир қанча қисм программалар ёрдамида қадамма қадам ҳисобланадиган мураккаб мисолларни ечишда қўл келади. Бундай командаларга бўлаклаб интеграллаш командаси **inparts** ва ўзгарувчини алмаштириш командаси **changevar** киради.

Бўлаклаб интеграллаш формуласи:

$$\int u(x)v'(x)dx = u(x)v(x) - \int u'(x)v(x)dx$$

Агар интеграл оситдаги функцияни $f=u(x)v'(x)$ деб олсак у ҳолда бўлаклаб интеграллаш командасининг параметрлари қуйидагича бўлади: **intparts(Int(f, x), u)**, бу ерда u – бошланғич функцияси бўлаклаб интеграллаш формуласи орқали топиладиган **u(x)** функция.

Агар интегралда $x=g(t)$ ёки $t=h(x)$ ўзгарувчиларни алмаштириш талаб қилинса, у ҳолда команданинг ўзгарувчиларни алмаштириш параметрлари қуйидагича бўлади: **changevar(h(x)=t,Int(f,x),t)** бу ерда t – янги ўзгарувчи.

Иккала **intparts** ва **changevar** командалари ҳам интегрални охиригача ҳисобламайди фақатгина оралиқ ҳисоблашларни бажаради. Охирги натижани олиш учун юқоридаги командаларни бажаргандан сўнг **value(%)** командасини киритиш керак. Бу ерда %–олдинги команданинг натижасини билдиради.

Бу командалардан фойдаланишдан олдин **with(student)** командаси ёрдамида **student** пакетини юклаш керак.

Назорат саволлари

1. Функциянинг ҳосиласи қайси команда ёрдамида топилади?
2. Юқори тартибли ҳосилаларни ҳисоблаш учун *dsolve* командасига қандай параметр киритилади?
3. Параметрга боғлиқ интегрални ҳисоблашда параметрларга чекланишлар қайси команда ёрдамида киритилади?
4. Бўлаклаб интеграллаш командасини тавсифланг.
5. Ўзгарувчиларни алмаштириш усули ёрдамида интеграллаш командасини тавсифланг.

6-мавзу: Maple тизимнинг график структуралари

Режа:

1. Икки ўлчовли графиклар
2. Уч ўлчовли графиклар
3. Анимация

1. Икки ўлчовли графиклар

plot командаси ва унинг параметрлари. Бир ўзгарувчи $f(x)$ функциянинг графигини чизиш учун (Ox ўқи бўйича $a \leq x \leq b$ интервалда ва Oy ўқи бўйича $c \leq y \leq d$ интервалда) қуйидаги команда ишлатилади:

plot(f(x), x=a..b, y=c..d, parameters)

бу ерда **parameters** – чизмани бошқариш параметри. Агар бу параметрлар берилмаса, у ҳолда жимлик қоидаси бўйича ўрнатилган параметрлар қабул қилинади. Инструментлар панелидан фойдаланган ҳолда чизманинг ҳолатини белгилаш мумкин.

plot командасининг асосий параметрлари:

1) **title="text"**, бу ерда **text** – чизманинг сарлавҳаси (матнни қўштирноқсиз ҳам ёзиш мумкин, агарда у фақатгина лотин ҳарфларидан иборат ва пробелсиз бўлса).

2) **coords=polar** – майдон координаталарининг қўйилиши (жимлик қоидаси бўйича декарт координаталар қўйилади).

3) **axes** – координата ўқлари турининг қўйилиши: **axes=NORMAL** – оддий ўқлар; **axes=BOXED** – график масштаби рамкада; **axes=FRAME** – маркази расмнинг чап томондан пастда жойлашган ўқлар; **axes=NONE** – ўқларсиз.

4) **scaling** – расм масштабининг қўйилиши: **scaling=CONSTRAINED** – ўқлар бўйича бир хил масштабда; **scaling=UNCONSTRAINED** – ойнанинг ўлчамига қараб график масштабланади.

5) **style=LINE(POINT)** – чизикларда чиқариш (ёки нуқталарда).

6) **numpoints=n** – графикнинг аниқланган нуқталар сони (жимлик қоидаси бўйича $n=49$).

7) **color** – чизик рангининг қўйилиши: рангларнинг инглизча талқини, масалан, **yellow** – сариқ ва ҳ.к.

8) **xtickmarks=nx** ва **ytickmarks=ny** – Ox ва Oy ўқлари-да белгилашлар сони.

9) **thickness=n**, бу ерда $n=1,2,3,\dots$ – чизик қалинлиги (жимлик қоидаси бўйича $n=1$).

10) **linestyle=n** – чизик тури: узлуксиз, пунктирли ва ҳ.к. ($n=1$ – узлуксиз, жимлик қоидаси бўйича ўрнатилган).

11) **symbol=s** – символ тури, у қуйидаги нуқталарни ўз ичига олади: **BOX**, **CROSS**, **CIRCLE**, **POINT**, **DIAMOND**.

12) **font=[f,style,size]** – матнни киритиш учун шрифт турини танлаш: **f** - шрифт номи: **TIMES**, **COURIER**, **HELVETICA**, **SYMBOL**; **style** – шрифт стили: **BOLD**, **ITALIC**, **UNDERLINE**; **size** – шрифт ўлчами pt. да.

13) **labels=[tx,ty]** – координата ўқлари бўйича ёзувлар: **tx** – Ox ўқи бўйича ва **ty** – Oy ўқи бўйича.

14) **discont=true** – чексиз узилишларни қуришни кўрсатиш.

plot командаси ёрдамида $y=f(x)$ аниқ функциянинг графикларини, шу билан бирга $y=y(t)$, $x=x(t)$ параметрик кўринишда берилган функциянинг графикларини яшаш мумкин. Бу ҳолда команда қуйидагича киритилади:

`plot([y=y(t), x=x(t), t=a..b], parameters).`

Ошкормас функциянинг графигини яшаш. Агар функция $F(x, y) = 0$ кўринишда берилган бўлса, у ҳолда ушбу функция ошкормас функция ҳисобланади.

Ошкормас функциянинг графигини яшаш учун **plots** графикли пакетдан **implicitplot** командаси ишлатилади: **implicitplot(F(x,y)=0,x=x1..x2,y=y1..y2).**

Расмда матнли тушунтиришларни чиқариш. **plots** пакетида матнли тушунтиришларни расмга чиқариш учун **textplot** командаси мавжуд:

textplot([xo,yo,'text'], ptions), бу ерда **xo**, **yo** – **'text'** матнни киритишда бошланғич нуқталарининг координата-лари.

Бир нечта графикли объектларнинг битта расмда чиқарилиши. Кўп ҳолларда битта расмда ҳар хил командалар ёрдамида ҳосил бўлган бир нечта графикли объектларни киритишга тўғри келади, масалан, **plot** командаси ёрдамида ясалган графикни чиқариш ҳамда **textplot** командаси ёрдамида ҳосил бўлган матнли ёзувларни қўшиш. Бунинг учун командаларнинг натижалари ўзгарувчига таъминланади:

> p:=plot(...): t:=textplot(...):

Бунда экранда ҳеч нарса чиқмайди. Графикли чизмаларни экранга чиқариш учун **plots** пакетидан қуйидаги командани бажариш керак:

> with(plots): display([p,t], options).

Тенгсизликлар билан берилган икки ўлчовли соҳанинг ясалиши.

Агар $f_1(x, y) > c_1, f_2(x, y) > c_2, \dots, f_n(x, y) > c_n$ тенгсизлик-лар билан берилган икки ўлчовли соҳани ясаш керак бўлса, у ҳолда бунинг учун **plots** пакетидан **inequal** командасини ишлатиш мумкин.

inequal({f1(x,y)>c1,...,fn(x,y)>cn}, x=x1...x2, y=y1..y2, options) команданинг фигурали қавсларида соҳани аниқлайдиган тенгсизликлар системаси берилади, кейин эса координата ўқларининг ўлчамлари ҳамда параметрлари кўрсатилади. Параметрлар очик ҳамда ёпиқ чегараларнинг рангини, ички ва ташқи майдон ранглари, чегараларни кўрсатувчи чизик қалинлигини бошқаради:

- **optionsfeasible=(color=red)** – ички соҳа рангининг ўрнатилиши;
- **optionsexcluded=(color=yellow)** – ташқи соҳа рангининг ўрнатилиши;
- **optionsopen(color=blue, thickness=2)** – очик чегараларнинг чизик ранглари ҳамда қалинликларининг ўрнатилиши;
- **optionsclosed(color=green,thickness=3)** – ёпиқ чегараларнинг чизик ранглари ҳамда қалинликларининг ўрнатилиши.

2. Уч ўлчовли графиклар

Ошкор функция билан берилган сиртнинг графиги.

$z = f(x, y)$ функциянинг графигини **plot3d(f(x,y), x=x1...x2, y=y1...y2, options)** командадан фойдаланиб чизиш мумкин. Ушбу команданинг параметрлари **plot** командасининг параметрларига ўхшаш. **plot3d** командасининг кўп қўлланиладиган параметрларига қуйидаги параметр киради: **light=[angl1, angl2, c1, c2, c3]** – (**angl1, angl2**) координатали сферик нуқталардаги ёруғлик манбаи ҳосил қиладиган сирт ёритилганлиги. Ёруғликнинг ранги қиймати $[0,1]$ орлиқда бўлган сариқ (**c1**), яшил (**c2**) ва кўк (**c3**) рангларнинг миқдорлари билан аниқланади. **style=opt** параметр расмнинг стилини аниқлайди: **POINT** – нуқталар, **LINE** – чизиқлар, **HIDDEN** – кўринмас чи-зиқларни олиб ташлашдан ҳосил бўлган тўр, **PATCH** – тўлдирувчи (жимлик қоидаси бўйича ўрнатилган), **WIREFRAME** – кўринмас чизиқларни киритишдан ҳосил бўлган тўр, **CONTOUR** – даража чизиқлари, **PATCHCONTOUR** – тўлдирувчи ва даража чизиқлари. **shading=opt** параметр тўлдирувчининг интенсивлик функцияси беради, унинг қиймати жимлик қоидаси бўйича хуз га тенг, **NONE** – график рангланмасдан чиқади.

Параметрик функция билан берилган сирт графиги. Агар $x=x(u,v)$, $y=y(u,v)$, $z=z(u,v)$ параметрик кўринишда берилган функциянинг сирт графигини яшаш талаб қилинаётган бўлса, у ҳолда бу функциялар командада квадрат кавс ичида саналади: **plot3d([x(u,v), y(u,v), z(u,v)], u=u1..u2, v=v1..v2).**

Ошкормас функция билан берилган сирт графиги. $F(x, y, z) = c$, тенглама билан ошкормас ҳолда берилган сиртнинг уч ўлчовли графиги **plot** пакетининг қуйидаги командаси ёрдамида ясалади:

implicitplot3d(F(x,y,z)=c, x=x1..x2, y=y1..y2, z=z1..z2),

бу ерда $F(x, y, z) = c$ сирт тенгламаси ва расмнинг координата ўқлари бўйича ўлчамлари берилади.

Фазовий эгри чизиқларнинг графиги. $x = x(t)$, $y = y(t)$, $z = z(t)$ параметрик кўринишда берилган фазовий эгри чизиқларни яшаш учун **plot**

пакетида **spacecurve** командаси мавжуд ва бу команданинг параметрлари қуйидагидан иборат:

> **spacecurve**([**x(t),y(t),z(t)**],**t=t1..t2**),

бу ерда **t** ўзгарувчи **t1** дан **t2** гача ўзгаради.

3. Анимация

Maple ҳаракатланувчи тасвирларни экранда **plot** пакетининг **animate** (икки ўлчовли) ва **animate3d** (уч ўлчовли) командалари ёрдамида чиқариш имконини беради. **animate3d** команданинг параметрлари ичида **frames** параметри мавжуд бўлиб, у анимациянинг кадрлар сонини аниқлайди (жимлик қондаси бўйича **frames=8**).

Уч ўлчовли тасвирларнинг параметрларини **plot3d** командаси ёрдамида эмас, балки дастурнинг контекстли менюсидан фойдаланиб созлаш мақсадга мувофиқдир. Бунинг учун тасвир устига сичқончани йўналтириб ўнг тугмасини босиш керак. Натижада тасвир параметрларини созлаш контекстли менюси пайдо бўлади. Ушбу менюнинг командалари тасвир рангини ўзгартириш, ёритувчанлик режимини ўзгартириш, керакли ўқлар турини ўрнатиш, чизик тури ва ҳаракатланувчи тасвирларни бошқариш имконини беради.

Тасвир параметрларини созлашнинг контекстли менюси:



Назорат саволлари

1. Текисликда ва фазода графиклар қайси командалар ёрдамида қурилади?
2. Қўшимча график командалар пакети қандай номланади?
3. Ошкормас функциянинг графиги қайси команда ёрдамида ҳосил қилинади?

4. *display* командасининг вазифаси.
5. *Тенгсизликлар системаси билан берилган икки ўлчамли соҳа қайси команда ёрдамида ҳосил қилинади?*
6. *Фазовий эгри чизиқлар ҳосил қилувчи командани тавсифланг.*
7. *animate ва animate3d командалари қандай имкониятларга эга?*

7-мавзу: Лимитларни ҳисоблаш. Функцияларни текшириш

Режа:

1. Лимитларни ҳисоблаш
2. Функцияларни текшириш

1. Лимитларни ҳисоблаш

Лимитни ҳисоблаш командасининг икки хил кўриниши мавжуд:

1) Тўғридан тўғри ҳисоблайдиган команда – **limit(expr, x=a, par)** бу ерда **expr** – ифода, **a**-х интилувчи нуқта, **par** – бир томонлама лимитни ҳисобловчи параметр (**left**–чапдан, **right**–ўнгдан) ёки ўзгарувчининг турини аниқловчи параметр (**real** – ҳақиқий, **complex** – комплекс).

2) Команданинг пассив ҳолати – **Limit(expr,x=a,par)** лимитнинг математик талқиндаги ёзувини чиқариб беради. Бу ерда ҳам параметрлар юқоридагидек. Мисол:

> Limit(sin(2*x)/x,x=0);

$$\lim_{x \rightarrow 0} \frac{\sin(2x)}{x}$$

> limit(sin(2*x)/x,x=0);

2

Бу икки команда математик ифодаларни стандарт аналитик кўринишда ёзиш имконини беради. Масалан:

> Limit(x*(Pi/2+arctan(x)),x=-infinity)=

limit(x*(Pi/2+arctan(x)), x=-infinity);

$$\lim_{x \rightarrow \infty} x \left(\frac{\pi}{2} + \arctan(x) \right) = -1$$

Бир томонлама лимитлар **left** – чап томондан ва **right** – ўнг тамондан параметрларини киритиш орқали ҳисобланади. Масалан:

> Limit(1/(1+exp(1/x)),x=0,left)=

limit(1/(1+exp(1/x)),x=0,left);

$$\lim_{x \rightarrow 0^-} \frac{1}{1 + e^{\frac{1}{x}}} = 1$$

> Limit(1/(1+exp(1/x)),x=0,right)=

limit(1/(1+exp(1/x)), x=0,right);

$$\lim_{x \rightarrow 0^+} \frac{1}{1 + e^{\frac{1}{x}}} = 0$$

2. Функцияни текшириш

Функцияни текширишни унинг аниқланиш соҳасини топишдан бошлаш керак. Лекин бу амални автоматлаштириш қийин. Шунинг учун бу масалани қараётганда тенгсизликни ечишга тўғри келади. Бироқ функция бутун сонлар ўқида аниқланганми ёки йўқми деган саволга уни узлуксизликка текшириш орқали жавоб бериш мумкин.

Функция узлуксизлиги ва узилиш нуқталари. $f(x)$ функциянинг берилган $[x_1, x_2]$ ораликда узлуксизлигини **iscont(f,x=x1..x2)** командаси орқали текшириш мумкин. Агар f функция бу ораликда узлуксиз бўлса у ҳолда чиқариш майдонида *true* (рост) жавоби пайдо бўлади, агар f функция бу ораликда узлуксиз бўлмаса у ҳолда чиқариш майдонида *false* (ёлғон) жавоби пайдо бўлади. Хусусий ҳолда **x=-infinity..+infinity** оралиқ берилса у ҳолда f функция бутун сонлар ўқида текширилади. Бунда агар *true* жавоби олинса у ҳолда айтиш мумкинки функция бутун сонлар ўқида аниқланган ва узлуксиз, акс ҳолда узилиш нуқталарини аниқлашга тўғри келади. Буни икки усулда амалга ошириш мумкин:

1) **discont(f, x)** командаси ёрдамида, бу ерда f – узлуксизлиги текшириладиган функция, x – ўзгарувчи. Бу команда ёрдамида биринчи ва иккинчи тур узилиш нуқталарини топиш мумкин.

2) **singular(f, x)** командаси ёрдамида, бу ерда f – функция, x – ўзгарувчи. Бу команда ёрдамида ўзгарувчининг ҳақиқий ва комплекс қийматларида иккинчи тур узилиш нуқталарини топиш мумкин.

Бу командаларни ишлатишдан олдин уларни албатта **readlib(name)** командаси ёрдамида стандарт кутубхонадан юклаб олиш керак. Бу ерда **name** – юқорида кўрсатилган ихтиёрий командалардан бирининг номи.

Бу иккала командада ҳам узилиш нуқталари фигурали қавсда кетма – кет кўринишда чиқарилади. Бундай ёзув **set(тўплам)** деб аталади. Узилиш нуқталарининг олинган қийматларини кейинчалик ишлатиш мумкин бўлиши учун **set** турини **convert** командаси ёрдамида оддий сонли турга ўтказиш керак.

Мисол: $y = e^{\frac{1}{x+3}}$ функциянинг узилиш нуқталарини топинг.

```
> readlib(iscont): readlib(discont):
```

```
> iscont(exp(1/(x+3)),x=-infinity..+infinity);
```

false

Бу функциянинг узлуксиз эмаслигини билдиради. Шунинг учун қуйидаги команда ёрдамида функциянинг узилиш нуқталарини топиш керак:

```
> discont(exp(1/(x+3)),x);
```

{-3}

Экстремумлар. Функциянинг энг катта ва энг кичик қийматлари.

Maple да функцияни экстремумликка текшириш учун **extrema(f,{cond},x,'s')**, командасидан фойдаланилади. Бу ерда f - экстремумлари изланаётган функция, **{cond}** фигурали қавсда ўзгарувчи учун чекланишлар кўрсатилади, x – ўзгарувчи номи, **'s'** апострофларда экстремум нуқтасининг координатаси таъминланадиган ўзгарувчининг номи кўрсатилади. Агар **{}** фигурали қавс бўш қолдирилса экстремумларни излаш бутун сонлар ўқи бўйича амалга оширилади. Бу команданинг натижаси **set** турига тегишли бўлади. Масалан:

```
> readlib(extrema):
```

```
> extrema(arctan(x)-ln(1+x^2)/2, {}, x, 'x0');x0;
```

$$\left\{\frac{\pi}{4} - \frac{1}{2}\ln(2)\right\}$$

$$\{\{x=1\}\}$$

Биринчи сатрда функциянинг экстремуми, иккинчи сатрда экстремум нуқтаси келтирилган.

Афсуски бу команда қайси экстремум нуқтаси минимум қайсиси максимум деган саволга жавоб бера олмайди. $f(x)$ функциянинг x ўзгарувчи бўйича $x \in [x_1, x_2]$ ораликдаги максимумини топиш учун **maximize(f,x,x=x1..x2)** командасидан, минимумини топиш учун **minimize(f, x, x=x1..x2)** командасидан фойдаланилади. Агар ўзгарувчидан кейин 'infinity' ёки $x=-\text{infinity}..\text{infinity}$ оралиқ кўрсатилса у ҳолда **maximize** ва **minimize** командалари мос равишда максимум ва минимумларни бутун сонлар ўқидан ҳам ҳақиқий сонлар тўпламида ҳам комплекс сонлар тўпламида излайди. Агар бундай параметрлар кўрсатилмаса у ҳолда максимум ва минимумларни излаш фақат ҳақиқий сонлар тўпламида амалга оширилади. Масалан:

```
> maximize(exp(-x^2),{x});
```

1

Бу командаларнинг камчилиги шуки, улар фақат максимум ва минимум нуқталардаги функция қийматларини чиқаради. Шунинг учун $y=f(x)$ функцияни экстремумга текшириш масаласини тўлиқ ечиш, яъни экстремум нуқталарнинг характери (max ёки min) ва уларнинг (x, y) координаталарни аниқлаш учун аввал:

```
> extrema(f,{x},s');s;
```

командасини, сўгра эса **maximize(f,x); minimize(f,x)** командаларини бажариш керак. Шундан кейингина барча экстремумларнинг координаталари тўлиқ топилади ва уларнинг характеристикалари (max ёки min) аниқланади.

maximize ва **minimize** командалари абсолют экстремумларни тез топади, аммо ҳар доим ҳам локал экстремумларни топишга қодир эмас. **extrema** командаси функция экстремумга эга бўлмаган критик нуқталарни ҳам ҳисоблайди. Бу ҳолда биринчи чиқариш сатрида функциянинг экстремал қийматлари иккинчи чиқариш сатридаги ҳисобланган критик нуқталардан кам бўлади. $f(x)$ функциянинг $x=x_0$ нуқтадаги топилган экстремумининг

характерини аниқлаш учун унинг шу нуқтадаги иккинчи тартибли ҳосиласини ҳисоблаш ва унинг ишорасига қараб хулоса қилиш мумкин. Агар $f''(x_0) > 0$ бўлса у ҳолда x_0 да $\min$ бўлади, агар $f''(x_0) < 0$ бўлса у ҳолда $\max$ бўлади.

Maple пакетининг *Maple* 6 версиясидан бошлаб **maximize** ва **minimize** командаларининг юқоридаги тавсифланган камчиликлари бартараф этилган. Максимум ёки минимум нуқталарининг координаталарини олиш учун бу командаларнинг параметрларида ўзгарувчидан кейин вергул билан янги **location** опциясини ёзиш керак. Натижада чиқариш сатрида функция максимуми (минимуми)дан кейин фигурали қавсда максимум (минимум) нуқталарнинг координаталари кўрсатилади. Масалан:

> minimize($x^4 - x^2$, x, location);

$$-\frac{1}{4}, \left\{ \left[\left\{ x = -\frac{1}{2}\sqrt{2} \right\}, -\frac{1}{4} \right], \left[\left\{ x = \frac{1}{2}\sqrt{2} \right\}, -\frac{1}{4} \right] \right\}$$

Чиқариш сатрида минимумларнинг координаталари ва бу нуқталарда функциянинг қийматлари ҳосил бўлди.

extrema, **maximize** ва **minimize** командалари албатта стандарт кутубхонадан **readlib(name)** командаси ёрдамида юклаб олиниши керак. Бу ерда **name** — юкланадиган команданинг номи.

Функцияни умумий схема бўйича текшириш.

1. $f(x)$ функциянинг аниқланиш соҳасини функцияни узлуксизликка текширгандан сўнг тўлиқ ўрнатиш мумкин.

2. $f(x)$ функциянинг узлуксизлиги ва узилиш нуқталари қуйидаги схема бўйича текширилади:

> iscont(f, x=-infinity..infinity);

> d1:=discont(f,x);

> d2:=singular(f,x);

Натижада $d1$ ва $d2$ ўзгарувчилар тўпламига 1 ва 2-тур узилиш нуқталаридаги x -координаталарнинг қийматлари таъминланади.

3. Асимптоталар. Чексиз узилиш нуқталари $f(x)$ функция графигининг вертикал асимптоталарини аниқлайди. Вертикал асимптотанинг тенгламаси қуйидаги кўринишга эга бўлади:

> ут:=d2;

$f(x)$ функциянинг чексизликдаги ҳолати оғма асимптоталари (агар улар мавжуд бўлса) билан характерланади. Оғма асимптотанинг тенгламаси $y=kx+b$, бу ерда коэффицентлар қуйидаги формулалар ёрдамида ҳисобланади:

$$k = \lim_{x \rightarrow +\infty} \frac{f(x)}{x} \text{ ва } b = \lim_{x \rightarrow +\infty} (f(x) - kx).$$

$x \rightarrow -\infty$ учун формулалар бир хил. Шунинг учун оғма асимптоталарни топишни қуйидаги схема бўйича амалга ошириш мумкин:

> k1:=limit(f(x)/x, x=+infinity);

> b1:=limit(f(x)-k1*x, x=+infinity);

> k2:=limit(f(x)/x, x=-infinity);

> b2:=limit(f(x)-k2*x, x=-infinity);

Кўпиннча $k1=k2$ ва $b1=b2$ бўлади. Бу ҳолда $x \rightarrow +\infty$ ва $x \rightarrow -\infty$ учун битта асимптота бўлади. Буни ҳисобга олган ҳолда асимптота тенгламаси тузилади

> уп:=k1*x+b1;

4. Экстремумлар. $f(x)$ функцияни экстремумга текширишни қуйидаги схема бўйича амалга ошириш мумкин:

> extrema(f(x), {}, x, 's');

> s;

> fmax:=maximize(f(x), x);

> fmin:=minimize(f(x), x);

Бу командаларни бажаргандан сўнг $f(x)$ функция барча максимум ва минимумларининг (x, y) координаталари топилади.

Функция графигини куриш. $f(x)$ функциянинг графигини куриш функцияни текширишнинг якуний босқичи. Расмда текширилаётган функция

графикдан ташқари унинг барча асимптоталари пунктир чизиқлар билан кўрсатилиши, $\max$ ва $\min$ нукталарнинг координаталари ёзилиши керак. Буни қуйидаги мисолда кўрамыз:

$y = \arctg(x^2)$ функциянинг графикини ва унинг асимптотасини қуриш, экстремум нукталарининг координаталарини кўрсатиш талаб қилинсин. Стандарт кутубхонадан барча керакли командаларни юклаймиз.

```
> restart: y:=arctan(x^2):
```

```
> iscont(y, x=-infinity..infinity);
```

true

```
> k1:=limit(y/x, x=-infinity);
```

k1:=0

```
> k2:=limit(y/x, x=+infinity);
```

k2:=0

```
> b1:=limit(y-k1*x, x=-infinity);
```

b1:= $\frac{1}{2}\pi$

```
> b2:=limit(y-k1*x, x=+infinity);
```

b2:= $\frac{1}{2}\pi$

```
> yh:=b1;
```

yh:= $\frac{1}{2}\pi$

```
> extrema(y,{x},s);
```

{0}

{ {x=0} }

```
> ymax:=maximize(y,{x}); ymin:=minimize(y,{x});
```

ymax:=

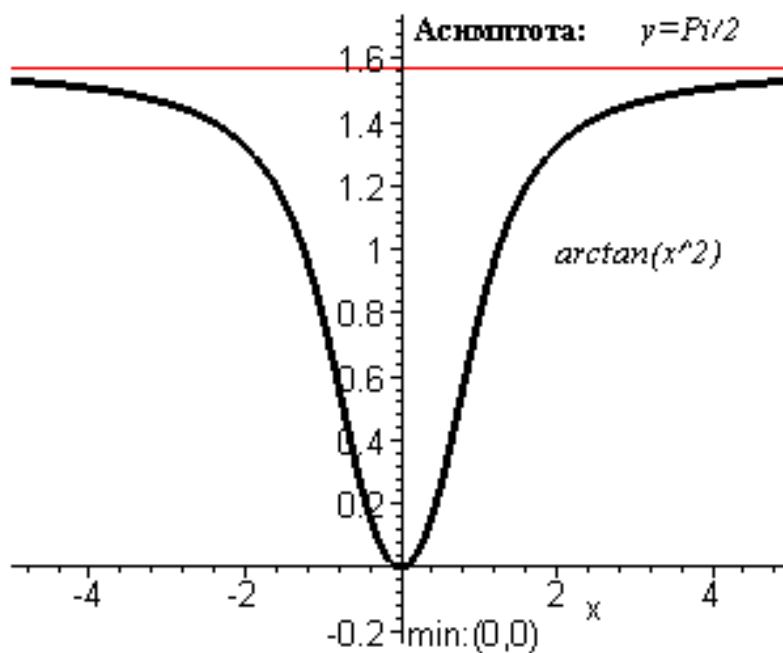
ymin:= 0

```
> with(plots): yy:=convert(y,string):
```

```

> p1:=plot(y,x=-5..5, linestyle=1, thickness=3,
color=BLACK):
> p2:=plot(yh,x=-5..5, linestyle=1,thickness=1):
> t1:=textplot([0.2,1.7,"Асимптота:"],
font=[TIMES, BOLD, 10], align=RIGHT):
> t2:=textplot([3.1,1.7,"y=Pi/2"],
font=[TIMES, ITALIC, 10], align=RIGHT):
> t3:=textplot([0.1,-0.2,"min:(0,0)"],
align=RIGHT):
> t4:=textplot([2,1,yy], font=[TIMES, ITALIC,
10], align=RIGHT):
> display([p1,p2,t1,t2,t3,t4]);

```



Назорат саволлари

1. Лимитни ҳисоблаш командаси ва унинг параметрларини тавсифланг.
2. Функцияни узлуксизликка текшириш командаларини тавсифланг.
3. Функциянинг тах ва тпн ларини топиш учун қандай командалар кетма-кетлиги киритилади?

4. *maximize, minimize* ва *extrema* командаларининг камчиликлари нимада?
5. *Функцияни текиширишнинг умумий схемасини келтиринг.*

8-мавзу: Дифференциал тенгламалар

Режа:

1. Дифференциал тенгламаларнинг аналитик ечими
2. Дифференциал тенгламаларнинг сонли ечими

1. Дифференциал тенгламаларнинг аналитик ечими

Дифференциал тенгламаларнинг умумий ечими. *Maple* да дифференциал тенгламаларнинг аналитик ечимларини топиш учун қуйидаги команда ишлатилади:

dsolve(eq,var,options), бу ерда **eq** – дифференциал тенглама, **var** – номаълум функциялар, **options** – параметрлар. Параметрлар масаланинг ечилиш усулини кўрсатиши мумкин, масалан, жимлик қоидаси бўйича аналитик ечим қуйидагича изланади: **type=exact**. Дифференциал тенгламаларни тузишда ҳосилани ифодалаш учун **diff** командаси ишлатилади, масалан, $y''+y=x$ дифференциал тенглама қуйидаги кўринишда ёзилади: **diff(y(x),x\$2)+y(x)=x**.

Дифференциал тенгламаларнинг умумий ечимлари сони дифференциал тенгламанинг тартибига боғлиқ бўлган ихтиёрий ўзгармасларга боғлиқдир. *Maple* да бундай ўзгармаслар қоида бўйича **_C1**, **_C2**, ва ҳ.к.лар билан белгиланади.

Бир жинсли бўлмаган чизиқли дифференциал тенгламанинг умумий ечими ҳамма вақт шундай чиқариладики, ушбу ечимнинг структураси аниқ кўринади. Маълумки бир жинсли бўлмаган чизиқли дифференциал тенгламанинг умумий ечими унга мос келувчи бир жинсли дифференциал тенгламанинг умумий ечимлари ҳамда берилган бир жинсли бўлмаган дифференциал тенгламанинг хусусий ечими йиғиндисига тенг. Шунинг учун ҳам бир жинсли бўлмаган чизиқли дифференциал тенгламанинг ечими ҳамма вақт ихтиёрий ўзгармасларни ўз ичига олган қўшилувчилардан (бу мос бир жинсли дифференциал тенгламанинг умумий ечими) ва ихтиёрий

ўзгармасларсиз қўшилувчилардан иборат (бу мазкур бир жинсли бўлмаган дифференциал тенгламанинг хусусий ечими) бўлиши мумкин.

dsolve командаси дифференциал тенгламанинг ечимини ҳисобланмайдиган шаклда беради. Ҳосил бўлган ечим устидан кейинчалик ишлаш учун (масалан, ечим графигини яшаш) ҳосил бўлган ечимнинг ўнг томонини **rhs(%)** команда билан ажратиш керак.

Ечимнинг фундаментал (базис) системаси. **dsolve** команда дифференциал тенглама ечимининг фундаментал системасини (базис функциялар) топиш имконини беради. Бунинг учун **dsolve** команданинг параметрларида **output=basis** ни кўрсатиш керак.

Мисол: $y^{(4)} + 2y'' + y = 0$ дифференциал тенглама ечимининг фундаментал системасини топинг:

```
> de:=diff(y(x),x$4)+2*diff(y(x),x$2)+y(x)=0;
```

$$de := \left(\frac{\partial^4}{\partial x^4} y(x) \right) + 2 \left(\frac{\partial^2}{\partial x^2} y(x) \right) + y(x) = 0$$

```
> dsolve(de, y(x), output=basis);
```

$$[\cos(x), \sin(x), x \cos(x), x \sin(x)]$$

Коши масаласи ёки чегаравий масаланинг ечими. Дифференциал тенглама билан бирга номаълум функциянинг бошланғич ёки чегаравий шартлари берилса **dsolve** командаси Коши масаласи ёки чегаравий масаланинг ечимини топиши мумкин. Бошланғич ёки чегаравий шартларда ҳосилаларни ифодалаш учун дифференциал оператор D ишлатилади, масалан, $y''(0)=2$ шарт $(D@@2)(y)(0)=2$ кўриниш берилади ёки $y'(1)=0$ шарт $D(y)(1)=0$ кўринишда берилади. n -чи тартибли ҳосила $(D@@n)(y)$ каби ёзилади.

Мисол: Коши масаласининг ечимини топинг: $y^{(4)} + y'' = 2\cos x$, $y(0) = -2$, $y'(0) = 1$, $y''(0) = 0$, $y'''(0) = 0$.

```
> de:=diff(y(x),x$4)+diff(y(x),x$2)=2*cos(x);
```

$$de := \left(\frac{\partial^4}{\partial x^4} y(x) \right) + \left(\frac{\partial^2}{\partial x^2} y(x) \right) = 2 \cos(x)$$

```
> cond:=y(0)=-2, D(y)(0)=1, (D@@2)(y)(0)=0,
```

```
(D@@3)(y)(0)=0;
```

```
cond:=y(0)=-2, D(y)(0)=1, (D(2))(y)(0)=0, (D(3))(y)(0)=0
```

```
> dsolve({de,cond},y(x));
```

$$y(x) = -2\cos(x) - x\sin(x) + x$$

Дифференциал тенгламалар системаси. Дифференциал тенгламалар системасининг (ёки Коши масаласининг) ечимини топиш учун **dsolve** командаси куйидаги кўринишда берилади:

dsolve({sys},{x(t),y(t),...}), бу ерда **sys** – дифференциал тенгламалар системаси, **x(t),y(t),...** – номаълум функциялар кетма – кетлиги.

Даражали қаторлар ёрдамида дифференциал тенгламаларни тақрибий ечиш. Кўпгина тур дифференциал тенгламаларнинг аниқ аналитик ечимини топиб бўлмайди. Бу ҳолда дифференциал тенгламаларнинг ечимини тақрибий усуллар ёрдамида, яъни номаълум функцияни даражали қаторга ёйиш орқали топиш мумкин.

Дифференциал тенгламанинг ечимини даражали қатор кўринишида топиш учун **dsolve** командада ўзгарувчилардан кейин **type=series** (ёки шунчаки **series**) параметрини кўрсатиш керак. **n** чи ёйилма тартибини яъни ёйишни нечанчи тартибгача бажариш кераклигини кўрсатиш учун, **dsolve** командадан олдин тартибни аниқлайдиган **Order:=n** командани кўйиш керак.

Агар дифференциал тенгламанинг умумий ечими даражали қаторлар ёйилмаси кўринишида изланаётган бўлса, у ҳолда топилган ёйилмадаги **x** нинг даражалари олдидаги коэффицентлар функциянинг нолдаги **y(0)** ва **D(y)(0)**, **(D@@2)(y)(0)** ва ҳ.к. ҳосилаларнинг номаълум қийматларини ўз ичига олади. Чиқариш сатрида ҳосил бўлган ифода, мазкур ечимни Маклорен қаторига

ёйгандаги кўринишда бўлади, аммо бунда x нинг даражалари олдидаги коэффициентлар бошқача бўлади. Хусусий ечимни ажратиш учун бошланғич $y(0)=y_1$, $D(y)(0)=y_2$, $(D@@2)(y)(0)=y_3$ ва ҳ.к. шартларни беришга тўғри келади. Ушбу бошланғич шартларнинг сони мос дифференциал тенгламанинг тартибига тўғри келиши керак.

Даражали каторларга бўлиш **series** турига эга бўлади, шунинг учун ушбу катор билан кейинчалик ишлаш учун **convert(%, polynom)** командаси ёрдамида уни кўпхадга айлантириб, ҳосил бўлган ифоданинг ўнг томонини **rhs(%)** команда ёрдами билан ажратиш керак.

2. Дифференциал тенгламаларни сонли ечиш

dsolve командаси ёрдамида дифференциал тенгла-маларни сонли ечиш. Дифференциал тенгламаларнинг сонли ечимини топиш учун (Коши масаласи ёки чегаравий масала) **dsolve** командасида **type=numeric** (ёки оддий **numeric**) параметрини кўрсатиш керак. У ҳолда дифференциал тенгламани ечиш команда қуйидаги кўринишга эга бўлади:

dsolve(eq, vars, type=numeric, options)

бу ерда **eq** – тенглама, **vars** – номаълум функциялар рўйхати, **options** – дифференциал тенгламани сонли интеграллаш усулини кўрсатувчи параметр. *Maple* да қуйидаги усуллар мавжуд:

method=rkf45 – 4-5-даражали Рунге-Кутта-Фельберг усули (жимлик қондаси бўйича шу усул ўрнатилган);

method=dverk78 – 7-8-даражали Рунге-Кутта усули;

method=classical – 3-даражали классик Рунге-Кутта усули;

method=gear ва **method=mgear** – Гирнинг бир қадамли ва кўп қадамли усуллари.

Дифференциал тенглама сонли ечимининг графигини **odeplot(dd, [x,y(x)], x=x1..x2)**, командаси ёрдамида куриш мумкин. Бу ерда функция сифатида **dd:=dsolve({eq,cond}, y(x), numeric)** дифференциал тенгламани сонли

ечиш командасидан фойдаланилади. Ундан сўнг квадрат қавс ичида ўзгарувчи ва номаълум функция $[x, y(x)]$, графикни куриш учун $x=x1..x2$ оралик кўрсатилади.

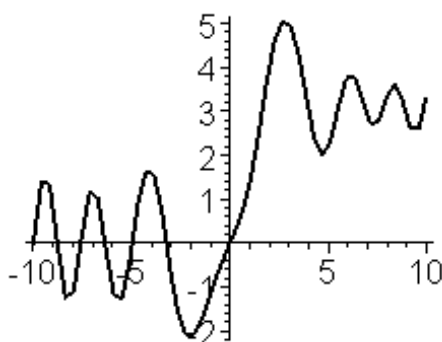
Мисол. $y' - x \sin(y) = \sin 2x$, $y(0) = 0$, $y'(0) = 1$ Коши масаласининг сонли ечимини топинг унинг графигини қуринг.

Бу мисолни ечиш учун қуйидаги командаларни киритамиз:

```
> restart; Ordev=6:
> eq:=diff(y(x),x$2)-x*sin(y(x))=sin(2*x):
> cond:=y(0)=0, D(y)(0)=1:
> de:=dsolve({eq,cond},y(x),numeric);
de:=proc(rkf45_x)...end
> de(0.5);
```

$$\left[x = .5, y(x) = .5449261153862630, \frac{\partial}{\partial x} y(x) = 1.272503082225380 \right]$$

```
> with(plots):
> odeplot(de,[x,y(x)],-10..10,thickness=2);
```



Назорат саволлари

1. Дифференциал тенгламаларни ечиш командаси ва унинг параметрларини тавсифланг.
2. Дифференциал тенгламаларда ҳосилалар қайси оператор ёрдамида ифодланади?
3. Дифференциал тенгламаларнинг фундаментал системасини ҳосил қилиш учун **dsolve** командасига қандай қўшимча параметрлар киритилади?

4. Дифференциал тенгламаларни сонли ечиш учун **dsolve** командасига қандай қўшимча параметрлар киритилади?
5. Дифференциал тенгламаларни сонли ечимининг графигини қуриш командасини тавсифланг.

9-мавзу: Кўп ўзгарувчили функцияларнинг дифференциал ва интеграл ҳисоби

Режа:

1. Кўп ўзгарувчили функцияларнинг дифференциал ҳисоби
2. Кўп ўзгарувчили функцияларнинг интеграл ҳисоби

1. Кўп ўзгарувчили функцияларнинг дифференциал ҳисоби

Maple да кўп ўзгарувчили функцияларнинг дифференциал ва интеграл ҳисобига оид масалаларнинг кўпчилиги бир ўзгарувчили функцияларда қўлланиладиган командалар ёрдамида қўшимча параметрларни киритиш билан ечилади.

Хусусий ҳосилалар. $f(x_1, \dots, x_m)$ функциянинг хусусий ҳосиласи **diff** командаси ёрдамида ҳисобланади. Бу ҳолда команда формати қуйидагича бўлади:

$$\text{diff}(f, x_1\$n_1, x_2\$n_2, \dots, x_m\$n_m)$$

бу ерда $x_1, \dots, x_m$ – дифференциаллаш бажариладиган ўзгарувчилар, $\$$ белгисидан кейин мос дифференциаллаш даражалари кўрсатилади, Масалан,

$\frac{\partial^2 f}{\partial x \partial y}$ хусусий ҳосила **diff(f,x,y)** кўринишда ёзилади.

Мисол. $f = \arctg \frac{x}{y}$ функциянинг $\frac{\partial f}{\partial x}$ ва $\frac{\partial f}{\partial y}$ ҳосилаларини топиш

талаб этилсин. Бунинг учун қуйидаги командаларни киритамиз:

> f:=arctan(x/y);

> Diff(f,x)=simplify(diff(f,x));

$$\frac{\partial}{\partial x} \arctan \frac{x}{y} = \frac{y}{x^2 + y^2}$$

> Diff(f,y)=simplify(diff(f,y));

$$\frac{\partial}{\partial y} \arctan \frac{x}{y} = -\frac{x}{x^2 + y^2}.$$

Кўп ўзгарувчили функцияларнинг локал ва шартли экстремумлари.

Функцияларни локал ва шартли экстремумларга текшириш учун стандарт кутубхонадаги **extrema(f,{cond},{x,y,...},'s')** командасидан фойдаланилади. Бу ерда **cond** – шартли экстремумни топиш учун тенгсизлик кўринишида ёзиладиган чекланишлар.

Чекланишлардан сўнг фигурали қавсда **f** функция боғлиқ бўлган барча ўзгарувчилар кўрсатилади, сўнгга қўштирноқ ичида экстремум нуқталарнинг координаталари таъминланадиган **s** ўзгарувчи номи ёзилади. Агар чекланишлар кўрсатилмаса локал экстремумлар изланади.

extrema командаси барча критик нуқталарни, яъни экстремум мавжуд бўлмаган нуқталарни ҳам чиқаради. Экстремумларни бермайдиган критик нуқталарни **subs** оператори ёрдамида тўғридан-тўғри функцияга қўйиб ажратиб олиш мумкин.

Бир ўзгарувчили функциялар сингари кўп ўзгарувчили функцияларнинг энг катта ва энг кичик қийматлари **maximize(f,{x1,...,xn},range)** ва **minimize(f,{x1,...,xn}, range)** командалари ёрдамида ҳисобланади. Бу ҳолда функциядан кейин фигурали қавсларда барча ўзгарувчиларни, сўнгга эса ҳар бир ўзгарувчи учун энг катта ва энг кичик қийматлар изланадиган оралиқларни кўрсатиш керак.

Мисол. $f(x, y) = 2x^4 + y^4 - x^2 - 2y^2$ функциянинг

экстремумларини топиш талаб этилсин. Ечим қуйидаги кўринишда бўлади:

```
> restart: readlib(extrema):
> f:=2*x^4+y^4-x^2-2*y^2:
> extrema(f,{},{x,y},'s');s;
```

$$\{0, \frac{-9}{8}\}$$

$$\left\{ \{x=0, y=0\}, \left\{x=\frac{1}{2}, y=0\right\}, \left\{x=-\frac{1}{2}, y=0\right\}, \{x=0, y=1\}, \{x=0, y=-1\}, \left\{x=\frac{1}{2}, y=1\right\}, \right.$$

$$\left. \left\{x=\frac{1}{2}, y=-1\right\}, \left\{x=-\frac{1}{2}, y=1\right\}, \left\{x=-\frac{1}{2}, y=-1\right\} \right\}$$

Бу функция иккита экстремумга эга бўлиб $f_{\max}=0$ ва $f_{\min}=-9/8$. Функция $(0,0)$ нуқтада максимумга эришади.

2. Кўп ўзгарувчили функцияларнинг интеграл ҳисоби

Maple да икки ва уч каррали интегралларни ҳисоблайдиган махсус командалар мавжуд бўлиб улар **student** кутубхонасида жойлашган.

$\iint_D f(x, y) dx dy$ кўринишидаги икки каррали интегралларни ҳисоблаш

учун **Doubleint(f(x, y), D)**, командасидан фойдаланилади. Бу ерда **D** – интеграллаш соҳаси бўлиб қуйидаги форматлардан бирида ёзилади:

- $x=x1..x2, y=y1..y2$, бу ерда $x1, x2, y1, y2$ сонлар интеграллашнинг тўртбурчак соҳасини беради;
- $x=f1(y)..f2(y), y=y1..y2$, бу ерда $f1(y), f2(y)$ – интеграллаш соҳасини ўнгдан ва чапдан $y1$ дан $y2$ гача чегараловчи чизиклар;
- $x=x1..x2, y=g1(x)..g2(x)$, бу ерда $g1(y), g2(y)$ – интеграллаш соҳасини қуйидан ва юқоридан $x1$ дан $x2$ гача чегараловчи чизиклар;

$\iiint_V f(x, y, z) dx dy dz$ кўринишидаги интегрални ҳисоблаш учун

Tripleint(f(x, y, z), x, y, z, V), командасидан фойдаланилади, бу ерда **V** – интеграллаш соҳаси.

Бу иккала команда ҳам натижани олиш учун **value(%)** командасидан фойдаланиш керак.

Такрорий интегралларни **int** командасини такрорлаш орқали ҳисоблаш

мумкин, масалан, $\int_0^2 dy \int_0^1 x^2 y^3 dx$ интеграл қуйидагича ҳисобланади:

`> int(int(x^2*y^3, x=0..1), y=0..2);`

$$\frac{4}{3}$$

Назорат саволлари

1. *Maple* да хусусий ҳосилалар қандай ҳисобланади.
2. Икки қаррали интегралларни ҳисоблаш командасини тавсифланг
3. Уч қаррали интегралларни ҳисоблаш командасини тавсифланг.
4. Функцияларнинг локал ва шартли экстремумларга текшириш учун қайси командадан фойдаланилади?
5. Такрорий интеграллар қандай ҳисобланади?

10-мавзу: Maple тизимида қаторлар устида амаллар

Режа:

1. Қаторлар йиғиндиси ва кўпайтмаларни ҳисоблаш
2. Функцияларни даражали қаторга ва Тейлор қаторига ёйиш.
3. Функцияни Фурье қаторига ёйиш

1. Қаторлар йиғиндиси ва кўпайтмаларни ҳисоблаш

$\sum_{n=a}^b S(n)$ кўринишидаги чекли ва чексиз йиғиндилар **sum** ва **Sum**

командаоари ёрдамида ҳисобланади. Бу командаларнинг параметрлари бир хил: **sum(expr, n=a..b)**, бу ерда **expr** – йиғинди индексига боғлиқ ифода, **a..b** – йиғинди индекси чегаралари бўлиб йиғиндининг **n=a** дан **n=b** гача ҳисобланишини билдиради.

Агар чексиз қаторнинг йиғиндисини ҳисоблаш талаб этилса юқори чегара сифатида **infinity** киритилади.

Худди шу тарзда $\prod_{n=a}^b P(n)$ кўпайтма **product(P(n),n=a..b)** ва **Product**

P(n),n=a..b) командалари ёрдамида ҳисобланади.

Мисол: Умумий ҳади $a_n = \frac{1}{(3n-2)(3n+1)}$ бўлган қаторнинг тўла

йиғиндисини ва N та ҳадининг йиғиндисини топиш талаб этилсин. Бунинг учун қуйидаги командаларни киритамиз:

```
> restart: a[n]:=1/((3*n-2)*(3*n+1));
```

$$a_n := \frac{1}{(3n-2)(3n+1)}$$

```
> S[N]:=Sum(a[n], n=1..N)=sum(a[n], n=1..N);
```

$$S_N := \sum_{n=1}^N \frac{1}{(3n-2)(3n+1)} = -\frac{1}{3} \frac{1}{3N+1} + \frac{1}{3}$$

```
> S:=limit(rhs(S[N]), N=+infinity);
```

$$S := \frac{1}{3}$$

2. Функцияларни даражали қаторга ва Тейлор қаторига ёйиш

$f(x)$ функцияни a нуқта атрофида

$$f(x) = C_0 + C_1(x-a) + C_2(x-a)^n + \dots + O(x^n)$$

даражали қаторга ёйиш **series(f(x), x=a, n)** командаси орқали амалга оширилади. Бу ерда a – нуқта, n – қатор ҳадлари сони.

Шунга ўхшаш **taylor(f(x), x=a, n)** командаси $f(x)$ функцияни $x=a$ нуқта атрофида $n-1$ тартибгача Тейлор формуласи бўйича қаторга ёяди.

series ва **taylor** командалари **series** туридаги натижаларни беради. Олинган натижалар билан кейинчалик ишлаш мумкин бўлиши учун уларни **convert(%,polynom)** командаси ёрдамида кўпхад кўринишига келтириш керак бўлади.

Кўп ўзгарувчили $f(x_1, \dots, x_n)$ функцияни $(x_1, \dots, x_n)$ ўзгарувчилар тўплами бўйича $(a_1, \dots, a_n)$ нуқталар атрофида Тейлор қаторига ёйиш учун **mtaylor(f(x), [x1,...,xn], n)** командасидан фойдаланилади. Бу команда стандарт кутубхонада жойлашганлиги учун уни ишлатишдан аввал **readlib(mtaylor)** чақирилиши керак.

Мисол. $f(x) = e^{-x} \sqrt{x+1}$ функцияни $x_0=0$ нуқта атрофида 5-тартибли даражали қаторга ёйиш талаб этилсин. Бунинг учун қуйидаги командани киритамиз:

```
> f(x)=series(exp(-x)*sqrt(x+1), x=0, 5);
```

$$f(x) = 1 - \frac{1}{2}x - \frac{1}{8}x^2 + \frac{13}{48}x^3 - \frac{79}{384}x^4 + O(x^5)$$

3. Функцияни Фурье қаторига ёйиш

Процедуралар яратиш. *Maple* да фойдаланувчи ўзининг процедураларини яратиш имконияти мавжуд. Процедура сарлавҳа қисмидан бошланади. Сарлавҳа процедура номи, таъминлаш оператори, **proc** калит сўзи,

ундан сўнг қавс ичида вергуллар билан ажратиб ёзилган формал параметрлардан иборат бўлади.

Процедуранинг ишлашида носозликлар юз бермаслиги учун процедура сарлавҳасида фақат процедура ичида ишлатиладиган ўзгарувчиларни тавсифлаш тавсия этилади. Бунинг учун **local** калит сўзи ёзилиб ундан кейин локал ўзгарувчилар вергул билан ажратиб кўрсатилади.

Сарлавҳадан кейин фойдаланувчи томонидан тузилган командалардан иборат процедура танаси ёзилади. Процедура танасидаги охирги команда процедуранинг охирги натижасини чиқаришни амалга оширади. Процедура **end** калит сўзи билан тугаши шарт.

Процедуранинг умумий кўриниши (стандарт синтаксис):

> **name:=proc(var1, var2, ...) local vloc1, vloc2,...;**

> **expr1;**

> **expr2;**

.....

> **exprn;**

> **end;**

Maple да функцияни Фурье қаторига ёядиган команда мавжуд эмас. Аммо Фурье қаторига ёядиган процедура яратиш мумкин. $f(x)$ $2l$ даврий функцияни $[x_1, x_2]$ ораликда Фурье қаторига ёйиш талаб этилсин. У ҳолда Фурье қатори қуйидаги кўринишга эга бўлади:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos \frac{k\pi x}{l} + \sum_{k=1}^{\infty} b_k \sin \frac{k\pi x}{l},$$

Бу ерда

$$l=(x_2-x_1)/2; a_0 = \frac{1}{l} \int_{x_1}^{x_2} f(x) dx; a_k = \frac{1}{l} \int_{x_1}^{x_2} f(x) \cos \frac{k\pi x}{l} dx;$$

$$b_k = \frac{1}{l} \int_{x_1}^{x_2} f(x) \sin \frac{k\pi x}{l} dx.$$

Фурье қаторининг n та ҳадини қуйидаги процедура ёрдамида ҳосил қилиш мумкин:

```
> fourierseries:=proc(f,x,x1,x2,n) local k, l, a, b, s;
> l:=(x2-x1)/2;
> a[0]:=int(f,x=x1..x2)/l;
> a[k]:=int(f*cos(k*Pi*x/l),x=x1..x2)/l;
> b[k]:=int(f*sin(k*Pi*x/l),x=x1..x2)/l;
> s:=a[0]/2+sum(a[k]*cos(k*Pi*x/l)+
b[k]*sin(k*Pi*x/l), k=1..n);
> end;
```

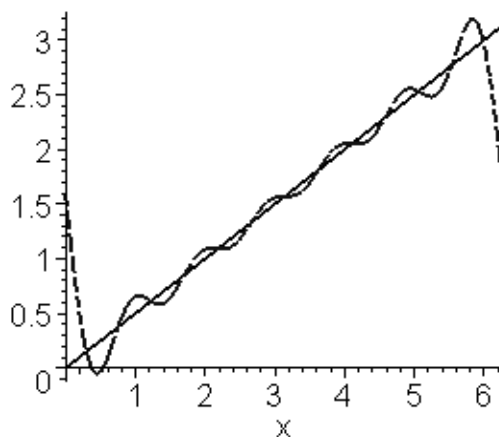
Бу процедурага қуйидаги мурожаат қилинади: **fourierseries(f,x,x1,x2,n)**, бу ерда **f** – қаторга ёйилиши керак бўлган функция номи, **x** – эркин ўзгарувчи, **x1, x2** – ёйиш оралиғи, **n** – қатор ҳадлари сони.

Мисол. $f(x)=x/2$ функцияни 2π давр билан $[0; 2\pi]$ ораликда 6-ҳадгача Фурье қаторига ёйиш, функция графигини қуриш ва Фурье қаторининг n та ҳади йиғиндисини топиш талаб қилинсин.

```
> f:=x/2:x1:=0:x2:=2*Pi;
> fr:=fourierseries(f,x,x1,x2,6);
```

$$fr := \frac{1}{2}\pi - \sin(x) - \frac{1}{2}\sin(2x) - \frac{1}{3}\sin(3x) - \\ - \frac{1}{4}\sin(4x) - \frac{1}{5}\sin(5x) - \frac{1}{6}\sin(6x)$$

```
> plot({fr,f}, x=x1..x2, color=[blue,black],
thickness=2, linestyle=[3,1]);
```



Пунктир чизик билан Фурье қаторининг n та ҳади йиғиндиси графиги, яхлит чизик билан ўша функциянинг графиги тасвирланган.

Назорат саволлари

1. *Maple да йиғинди ва қўпайтма қандай ҳисобланади?*
2. *Функцияларни даражаси қаторга ёйиш командасини тавсифланг.*
3. *Maple да процедуралар қандай яратилади*
4. *Қайси команда функцияларни Тейлор қаторга ёйиш учун ишлатилади?*
5. *Функцияларни Фурье қаторга ёйиш учун қандай командалар кетма-кетлигидан фойдаланилади?*

11-мавзу: Maple тизимининг дастурлаштириш воситалари

Режа:

1. Фойдаланувчи функциялари
2. Шарт оператори
3. Цикл операторлари
4. Процедур ва процедура-функциялар

1. Фойдаланувчи функциялари

Maple ядроси, кутубхона ва пакетлари 3000 дан ортиқ функцияларга эга бўлсада, тизимда фойдаланувчилар ўзларининг функцияларини яратиш имконияти мавжуд.

Функцияларни яратишнинг бир қанча усуллари мавжуд бўлиб энг оддий усул бирор бир ифода кўринишидаги функция қандайдир ўзгарувчига таъминланади: **name:=expr**, бу ерда **expr** – қандайдир ифода.

Бу усул аслида оддий таъминлаш операторини билдиради. Бу турдаги функциялар билан ишлашни қуйидаги мисол ёрдамида кўриб чиқамиз:

```
>m:=sqrt(x^2+y^2);
```

$$m := \sqrt{x^2 + y^2}$$

```
>x:=3: y:=4: m;
```

$$\sqrt{25}$$

```
>evalf(m);
```

5.00000000

Бу усулда берилган функциялар тўлақонли функция бўла олмайди, чунки уларда фақат глобал ўзгарувчилар ишлатилади ва функциянинг қиймати боғлиқ бўладиган параметрлар рўйхати кўрсатилмайди. Бу ҳолда функция ўзгарувчиларининг қийматини алоҳида таъминлаш оператори ёрдамида бериш керак бўлади.

Тўлақонли функцияларни яратиш функционал оператордан фойдаланишга асосланган. Бунинг учун қуйидаги конструкциядан фойдаланилади:

name:=(x,y,...)->expr

Функцияга мурожаат **name(x.y,...)**, кўринишда амалга оширилади. Бу ерда **(x.y,...)** – фойдаланувчи функциясининг формал параметрлар рўйхати. Формал параметрлар рўйхатида кўрсатилган ўзгарувчилар локал ўзгарувчилар ҳисобланади. Уларнинг ўрнига фактик параметрларни қўйганда улар бу қийматларни фақат функция танасида сақлаб қоладилар. Бу функциядан ташқарида бу ўзгарувчилар аниқланмаган бўлади ёки олдинги қийматини сақлаб қолади. Қуйида бу турдаги функцияларга доир мисоллар келтирилган:

>restart;

>x:=0; y:=0;

x:=0

y:=0

>m:=(x,y)->sqrt(x^2+y^2);

$$m := (x, y) \rightarrow \sqrt{x^2 + y^2}$$

>m(3, 4);

5

>m(3., 4);

5.000000000

>[x, y];

[0,0]

Бу мисолдан кўриниб турибдики $m(x, y)$ функциянинг қийматини ҳисоблаганда x ва y ўзгарувчилар 3 ва 4 қийматларни олади, лекин функциядан ташқарида улар олдинги қийматларини сақлаб қолади.

Фойдаланувчи функциясини яратишнинг яна бир усули қуйидаги команда асосида амалга оширилади:

name:=unapply(expr, var1, var2, ...)

Қуйида бу командани қўллашга доир мисол келтирилган:

```
>fm:=unapply(x^2+y^2, x, y);
```

$$fm := (x, y) \rightarrow x^2 + y^2$$

```
>fm(4, 3);
```

25

2. Шарт оператори

Тармоқланувчи дастурларни яратиш учун Maple да шарт оператори мавжуд бўлиб умумий ҳолда қуйидаги кўринишга эга:

```
if <Шарт> then <Элементлар>
|elif <Шарт> then <Элементлар>|
|else <Элементлар>|
fi:
```

Вертикал чизиқлар | | билан асосий бўлмаган элементлар кўрсатилган. Шарт операторининг қуйида келтирилган кўринишлари амалиётда кенг қўлланилади:

- **if <Шарт> then <1-элементлар> fi** – агар **Шарт** бажарилса, у ҳолда **1-элементлар** бажарилади, акс ҳолда ҳеч нарса бажарилмайди;
- **if <Шарт> then <1-элементлар> else <2-элементлар> fi** – агар **Шарт** бажарилса, у ҳолда **1-элементлар** бажарилади, акс ҳолда **2-элементлар** бажарилади;

Шарт сифатида (<, <=, >, >=, =, <>) таққослаш амаллари ва and, or, not мантиқий амаллар асосида тузилган ихтиёрий мантиқий ифода бўлиши мумкин.

Мисол:

```
> x:=-5;
```

```
> if x<0 then print('Negative') fi;
```

Negative

```
> x:=5;
```

```
> if x<0 then print('Negative') else print('Positive') fi;
```

Positive

3. Цикл операторлари

Баъзи ҳолларда қандайдир ифодани такрорий бажаришга тўғри келади. Бундай ҳолларда цикл операторидан фойдаланилади. Maple да цикл операторининг умумлашган шакли қуйидагича:

**| for <name>| |from <expr1>| |to <expr3>| |by <expr2>| (while <expr4>| do
<statement sequence> od;**

Бу ерда **name** – циклни бошқарувчи ўзгарувчи номи, **expr1**, **expr2**, **expr3** — name ўзгарувчининг бошланғич, охири ва ўзгариш қадамининг қийматини берадиган ифодалар, **expr4** – циклнинг бажарилишини белгилайдиган логик ифода, **statement sequence** – такрорланадиган ифода ёки операторлар тўплами.

Цикл бажарилиши давомида бошқарувчи ўзгарувчи **expr1** қийматдан **expr2** қийматгача **expr3** қадам билан ўзгаради. Агар **by <expr2>** қисми ёзилмаса бошқарувчи ўзгарувчи +1 қадам билан ўзгаради. Буни қуйидаги мисолда кўриш мумкин:

```
> for i from 1 to 5 do print(i) od;
```

```
1
2
3
4
5
```

Бошқарувчи ўзгарувчининг чегараларини арифметик ифодалар кўринишида ҳам бериш мумкин:

```
> for i from 7/(2+5) to 2+3 do print(i) od;
```

```
1
2
3
4
5
```

Қуйидаги мисолда бошқарувчи ўзгарувчининг қиймати 1 дан 10 гача 2 қадам билан ўзгаради:

```
> for i from 1 to 10 by 2 do printd) od:
```

```
1
3
5
7
9
```

Циклни қўшимча **while** <expr4> блоки ёрдамида тугатиш ҳам мумкин. Бундай ҳолда цикл **expr4** ифоданинг қиймати **false**(ёлғон) бўлгунча бажарилади:

```
> for i from 1 to 10 by 2 while i<6 do print(i) od:
```

```
1
3
5
```

Цикл операторнинг яна бир кўриниши қуйидагича:

|for <name>| |in <expr1>| |while <expr2>| do <statement sequence> od:

Бу ерда **expr1** - бошқарувчи ўзгарувчи қабул қиладиган қийматлар рўйхати. Бу ҳолда цикл рўйхатдаги қийматлар тугагунча ёки **expr2** ифода **false**(ёлғон) қиймат қабул қилгунча бажарилади. Масалан:

```
> for i in [1,2,5,-1,7,12] do print(i) od;
```

```
1
2
5
-1
7
12
```

```
> for i in [1,2,5,-1,7,12] while i>0 do print(i) od:
```

```
1
2
```

5

Баъзи ҳолларда цикл операторининг соддалаштирилган шаклидан фойдаланилади. Бундай ҳолда цикл оператори қуйидаги кўринишга эга бўлади:

while <expr> do <statement sequence> od:

Бу ерда цикл **expr** мантикий ифода **false**(ёлғон) қиймат қабул қилгунча бажарилади:

> n:=1;

n:=1

> while n<16 do 2*n od;

n:=2

n := 4

n := 8

n :=16

4. Процедура ва процедура-функциялар

Процедура – бу қисм дастур бўлиб у мустақил қийматга эга бўлади ва бир ёки бир нечта амалларни бажаради. Процедуралар структурали дастурлашнинг асосий элементи ҳисобланади ва *Maple* тизимининг имкониятларини кенгайтиришга хизмат қилади. Ҳар қандай процедура ягона номга ва параметрлар рўйхатига эга бўлади. Процедураларга мурожаат уларнинг номлари ва параметрлар рўйхатини келтириш орқали амалга оширилади. Бундай ҳолда одатда процедура бажарилгандан сўнг ҳеч қандай натижа бермайди. Процедура-функциялар эса уларга мурожаат қилганда қандайдир натижани беради. Юқорида тавсифланган фойдаланувчи функциялари процедура-функцияларнинг соддалаштирилган кўриниши ҳисобланади. Оддий ҳолда процедура қуйидаги кўринишда берилади::

name :=proc(Параметрлар)

Процедура танаси

end;

Процедура параметрлари уларнинг номларини санаш орқали кўрсатилади, масалан, **proc(x)** ёки **proc(x,y,z)**. Ўзгарувчи номидан кейин **::** белгиси орқали унинг турини аниқлаш мумкин, масалан, **proc(n::Integer)** да **n** ўзгарувчининг бутун турда эканлиги кўрсатилади. Процедурага мурожаат **name(параметрлар)** ифодаси орқали амалга оширилади.

Мисол сифатида комплекс соннинг модулини ҳисоблайдиган процедурани келтирамиз:

```
>modc:=proc(z)
>evalf(sqrt(Re(z)^2+Im(z)^2))
>end;
> modc(3.+I*4.);
```

5.000000000

Назорат саволлари

1. *Maple* да фойдаланувчи функциялари қандай яратилади?
2. Локал ўзгарувчиларни тавсифлаш учун қайси командадан фойдаланилади?
3. Шарт операторининг умумий кўриниши ва ишлаш принципини тавсифланг
4. Цикл операторларининг қандай шакллари мавжуд
5. Процедуралар қандай афзалликларга эга
6. Процедураларни яратиш жараёнини тушунтиринг
7. Процедура ва функцияларга мурожаат қандай амалга оширилади?

Фойдаланилган адабиётлар

1. С.Е. Савотченко, Т.Г. Кузьмичева Методы решения математических задач в Maple. Белгород. 2001 г.
2. Ёрунбаев Э., Муродов Ф. Компьютер алгебраси тизимларининг амалий тадбиқлари. Ўқув қўлланма:- СамДУ нашр матбаа-маркази, Самарқанд, 2003, 96 б.
3. Дьяконов В. Maple 6. Учебный курс СПб.: Питер, 2001. Введение в Maple.
4. Матросов А. Решение задач математики и механики в среде Maple 6. СПб.: Питер, 2000.

Мундарижа

Кириш	3
1- Мавзу: Компьютер алгебраси тизимлари тўғрисида умумий тушунчалар. Maple тизими график интерфейсининг таркибий қисмлари.....	5
2-Мавзу: Maple тизими командасининг синтаксиси. Математик ифодаларни қайта ишлаш командалари	13
3-мавзу: Тенглама ва тенгсизликларни ечиш командалари	27
4-мавзу: Чизиқли алгебра амалларини бажариш командалари..	34
5-мавзу: Дифференциаллаш ва интеграллаш командалари.....	48
6-мавзу: Maple тизимнинг график структуралари.....	55
7-мавзу: Лимитларни ҳисоблаш. Функцияларни текшириш.....	64
8-мавзу: Дифференциал тенгламалар.....	76
9-мавзу: Кўп ўзгарувчили функцияларнинг дифференциал ва интеграл ҳисоби.....	84
10-мавзу: Maple тизимида қаторлар устида амаллар.....	89
11-мавзу: Maple тизимининг дастурлаштириш воситалари.....	95
Фойдаланилган адабиётлар.....	105

