**Reprezentări grafice(cerc, triunghi)**

* Exemple in **reprez-geom.mw**

**Reprezentări grafice**

* Exemple in **grafice-exemple.mw**
* Principalele comenzi destinate reprezentărilor grafice sunt: **plot** şi **plot3d**.
* Acestora li se adaugă comenzile din pachete precum **plots** sau **plottools.**

Comanda ***plot*** poate fi folosită sub mai multe forme, cele mai importante fiind:

* **plot(f(x),x=a..b,opţiuni)** - realizează reprezentarea grafică a funcţiei f pe intervalul [a,b];
* **plot({f(x),g(x),……},x=a..b,opţiuni)** – realiazează reprezentarea grafică, în acelaşi sistem de axe, a funcţiilor f si g, pe intervalul [a,b]
* **plot([f(t),g(t),t=a..b],opţiuni)** – realizează reprezentarea grafică a curbei de ecuaţii parametrice x=f(t) şi y=g(t), unde parametrul t ia valori în intervalul [a,b]

La aceste sintaxe se pot adăuga ca parametric, opţiuni legate de culoare, stil, sistem de coordonate, titlu dat graficului.

Exemplu: plot(parametri\_de\_baza, color=….., title=….., coords=…….,title=”………..” );

Comanda ***plot3d*** are ca principale sintaxe:

* **plot3d(f(x,y),x=a..b,y=c..d,opţiuni)**- realizează reprezentarea grafică a unei funcţii f:[a,b]x[c,d]->R.
* **plot3d({f(x,y),g(x,y),…},x=a..b,y=c..d,optiuni)**- realizează reprezentarea grafică, în acelaşi sistem de axe, a funcţiilor f,g:[a,b]x[c,d]->R.
* **plot3d([f(u,v),g(u,v),h(u,v)],u=a..b,v=c..d,optiuni)**- realizează reprezentarea geometrică a suprafeţei de ecuaţii parametrice x=f(u,v),y=g(u,v),z=h(u,v), unde parametrii u şi v aparţin intervalelor [a,b], respectiv [c,d].

Şi în acest caz pot fi adăugate opţiuni de culoare, titlu, axe.

În ambele situaţii, opţiunile dorite pot fi selectate şi din meniul contextual aferent zonei 2Dgraphics, respectiv 3Dgraphics rezultată, meniu accesibil cu click-dreapta.

**Comenzi din pachetul plots**

* Exemple in **pachetul-plots.mw**
* **interactive(funcţie)** sau **interactive({listă funcţii})** ------ permite reprezentarea grafică interactivă a funcţiilor. Tipul graficului şi valorile variabilelor pot fi selectate dintr-o fereastră ce se deschide la apelul comenzii.
* Se încarcă pachetul plots**: with(plots)**
* Comenzile **animate** şi **animate3d** permit realizarea de animaţii în plan, respectiv spatiu prin reprezentarea grafică a unor funcţii parametrizate.

Sintaxe:

* **animate(comanda\_de\_reprezentare\_grafica(plot,plot3d), argumente\_pt\_comanda\_de reprezentare\_grafică, parametru=a..b(sau lista de valori L=[…]) , opţiuni)**
* **animate3d(f(x,y,t),x=a..b,y=c..d,t=t1..t2,opţiuni)**

Pentru obţinerea animaţiei se poate accesa opţiunea “animate-play” din meniul contextual al suprafeţei de desenare (click dreapta).

* **animatecurve(f(x),x=a..b)**  sau **animatecurve([x(t),y(t),t=a..b])** cu opţiunea play din meniul contextual permite vizualizarea desenării graficului funcţiei f pe intervalul [a,b] (o curbă).
* **display(lista/vector/multime grafice,optiuni)** are ca efect reprezentarea în acelaşi sistem de axe a graficelor date ca parametri. Un alias al acestei comenzi este şi **display3d**.
* **complexplot(f(x),x=a..b,optiuni) -**realizeazăreprezentarea grafică a unei funcţii complexe de variabilă reală f
* **complexplot([lista de numere complexe],optiuni) -**reprezintă grafic numerele complexe din lista dată ca parametru unite prin linii, în ordinea în care sunt scrise.
* **textplot({multime de puncte [x,y,”string”]},optiuni(aliniere,culori))** -permite reprezentarea unui şir de caractere la o anumită coordonată.

**Liste(lists)**

* Exemple in **sd.mw**

Listele sunt şiruri ordonate de expresii cuprinse între paranteze drepte, separate între ele prin virgulă.

Dacă L reprezinta o listă atunci L[i] desemnează elementul de pe poziţia i.

[] reprezintă lista vidă.

Operaţii de bază cu liste:

* **Adăugarea unui element x la o listă L:**

pe prima poziţie: [x,op(L)]

pe ultima poziţie: [op(L),x]

pe poziţia i: [op(1..i-1,L),x,op(i..-1,L)]

* Extragerea din lista L a elementelor de pe poziţiile de la i la j: L(i..j) sau [op(i..j,L)]
* Modificarea elementului de pe poziţia i din lista L: L[i]:=x sau subsop(i=x,L)
* Eliminarea elementului de pe poziţia i din lista L: subsop(i=NULL,L)
* Ordonarea elementelor unei liste: sort(L, tip\_sortare). Tipul sortării poate avea ca valori: ascending (ordine crescătoare), `>` (ordine descrescătoare), lexorder (ordinea lexicografică) , length (în funcţie de lungime).

**Comenzi din pachetul ListTools:**

* BinaryPlace(Lista\_ordonata L,x) returnează poziţia elementului după care urmeaza a fi adăugat elementul x pentru ca lista L să rămână ordonată.
* DotProduct(L1,L2) returnează produsul scalar al celor doua liste (care trebuie să aibă acelaşi număr de parametri).
* Interleave(L1,L2,....) interclasează două sau mai multe liste.
* Join(L,x) inserează elementul x între oricare două elemente consecutive ale listei L.
* MakeUnique(L) elimină duplicatele din lista L.
* Occurrences(x,L) returnează numărul de apariţii ale valorii x în lista L.
* PartialSums(L) returnează o listă L1 care are ca elemente sumele parţiale ale listei L (L1[i]= suma primelor i elemente din lista L).
* Reverse(L) returnează lista elementelor listei L în ordine inversă.
* Search(x,L) returnează poziţia primei apariţii a elementului x în lista L.
* SearchAll(x,L) returnează poziţiile tuturor apariţiilor elementului x în lista L.
* Sorted(L) verifică dacă lista L este ordonată crescător returnând o valoare booleană.

***Multimi(sets)***

Mulţimile sunt şiruri neordonate de expresii cuprinse între acolade, separate prin virgule.

{} reprezintă mulţimea vidă.

Duplicatele sunt automat eliminate, iar în zona de output mulţimile sunt ordonate.

Operaţii cu multimi:

* reuniunea: A ***union*** B
* intersecţia: A ***intersect*** B
* diferenţa: A ***minus*** B